Imitating Opponent to Win: Adversarial Policy Imitation Learning in Two-player Competitive Games

The Viet Bui
Singapore Management University
Singapore, Singapore
tvbui@smu.edu.sg

Tien Mai
Singapore Management University
Singapore, Singapore
atmai@smu.edu.sg

Thanh H. Nguyen
University of Oregon
Eugene, Oregon, United States
thanhhng@cs.uoregon.edu

ABSTRACT

Recent research on vulnerabilities of deep reinforcement learning (RL) has shown that adversarial policies can influence a target RL agent (victim agent) to perform poorly. In existing studies, adversarial policies are directly trained based on experiences of interacting with the victim agent. A key shortcoming of this approach is that knowledge derived from historical interactions may not be properly generalized to unexplored policy regions of the victim agent, making the trained adversarial policy significantly less effective. In this work, we design a new effective adversarial policy learning algorithm that overcomes this shortcoming. The core idea of our new algorithm is to create a new imitator — the imitator will learn to imitate the victim agent's policy while the adversarial policy will be trained based on both interactions with the victim agent and feedback from the imitator to forecast victim's intention. By doing so, we can leverage the capability of imitation learning in well capturing underlying characteristics of the victim policy. Our victim imitation learning model differs from prior models as the environment's dynamics are driven by adversary's policy and will keep changing during the adversarial policy training. We provide a provable bound to guarantee a desired imitating policy when the adversary's policy becomes stable. We further strengthen our adversarial policy learning by incorporating the opposite of the adversary's value function to the imitation objective, leading the imitator not only to learn the victim policy but also to be adversarial to the adversary. Finally, our extensive experiments using four competitive MuJoCo game environments show that our proposed algorithm outperforms state-of-the-art algorithms.[1]

KEYWORDS

Reinforcement Learning; Non-zero-sum Multi-agent Competition; Adversarial Policy; Imitation Learning

ACM Reference Format:
The Viet Bui, Tien Mai, and Thanh H. Nguyen. 2023. Imitating Opponent to Win: Adversarial Policy Imitation Learning in Two-player Competitive Games. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 16 pages.

1 INTRODUCTION

Exploring vulnerabilities of deep reinforcement learning has drawn a lot of interests from the AI research community [2, 17, 20, 21], given recent successes of deep RL in accomplishing a variety of interesting multi-agent learning tasks [3, 14, 18, 19]. Most of the existing work follows the traditional adversarial learning framework which often makes strong assumptions about the adversary's capabilities. Typically, the attacker is assumed to be able to manipulate input image observations or even interfere with the learning process of the victim agent. As pointed out in some recent work [7], these assumptions are not practical, especially in real-world domains such as autonomous driving in which the attacker cannot easily modify the input of the victim policy.

A recent alternative attack approach to the victim policy was introduced in [7] which presents the idea of *adversarial policy*. Essentially, an attacker can build an adversarial policy for an opponent agent (this agent is under control by the attacker, thus is called adversary agent in our paper) that takes actions in a shared environment with the victim. This adversarial policy can weaken the outcome of the victim's policy, not by making the opponent choose stronger actions, but instead by inducing natural observations that can lead the victim to behave in an undesired way. Motivated by these results on negative effects of adversarial policy on the victim's policy, our paper focuses on designing new stronger adversarial policies in the non-zero-sum Markov game setting.

Our key contribution is to introduce an imitator of which goal is to discover intrinsic properties of the victim's policy from historical interactions between the victim and adversary agents — this knowledge is then transferred to the attacker to improve the adversarial policy. By following this idea, we are able to exploit the advantage of imitation learning [9] to anticipate the victim agent's moves in unexplored policy regions, allowing us to strengthen the generated adversarial policy. This is a significant advancement compared to previous work [7, 8] which learns an adversarial policy directly from past interactions with the victim agent, substantially limiting the impact of the trained adversarial policy on unseen policy regions of the victim agent.

A key challenge in incorporating our new imitator into the adversarial policy learning framework is that the imitator is trained simultaneously with the adversarial policy learning — the inter-dependency between the imitator and the adversary agent complicates the entire training process. Note that in traditional imitation learning, the set of the victim's policy trajectories used for training the imitator is fixed during the training process. On the other hand, in the context of adversarial policy learning, the environment's dynamics (and as a result, the victim trajectories obtained for training the imitator) are driven by the adversarial policy. The adversarial policy, in turn, is trained based on the policy output of the imitator. This learning inter-dependency complication requires a carefully

[1] Codes and models are available at https://github.com/vietbt/APIL_CompetitiveGames.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

designed training process to ensure a convergence to high-quality outcomes for both the imitation policy and the adversarial policy.

To address this learning challenge, we provide the following contributions: (i) we theoretically characterize the dependency between the adversarial policy and the imitation policy; (ii) we provide a provable bound to guarantee a desired imitating policy when the adversary's policy becomes stable; and (iii) we further strengthen our adversarial policy learning by enhancing our imitator — we incorporate the negation of the adversary's goal into the imitator's objective function, making the imitator to both learn the victim policy and be adversarial to the adversary agent.

Lastly, we conduct extensive experiments on four competitive MuJoCo game environments introduced by Emergent Complexity [1] (including Kick And Defend, You Shall Not Pass, Sumo Humans, and Sumo Ants) to evaluate our proposed adversarial policy learning algorithms. We empirically show that our generated adversarial policies obtained a significantly higher winning (plus tie) rates against the victim agent in these game environments, in comparison with state-of-the-art adversarial policy methods. We further show that we can make the victim agent substantially more resilient to any adversarial policies (generated by different algorithms) by retraining the victim policy against our adversarial policies.

2 RELATED WORK

Attacks on deep RL. Existing works have focused on creating attacks by manipulating victim's observations or victim's actions. For example, [10] and [2] propose to perturb victim's observations to force the victim to make sub-optimal actions, thus fails the task. Later works [12, 16, 22, 26, 32] extend this approach by proposing to manipulate victim's observations at some selected time steps, instead of the whole trajectories. Other papers [2, 10, 15, 30, 33] focus on attacks through observation manipulation but in black-box settings, i.e., the adversary does not have the power to manipulate victim's observations, but can access the input and output of the victim's policy or deep-Q networks. Besides, there are works that propose to directly perturb actions taken by victim agents in both white-box and black-box settings [13, 30]. There is another emerging line of approaches that view the attacks as a two-player competitive game, i.e., focusing on training an adversarial agent to play with the victim. For example, [7] train an adversarial agent using Proximal Policy Optimization (PPO) [24] for a set of MuJoCo game environments [28], and [8] propose to break the zero-sum-game setting and redesign the adversary's reward function to achieve better adversary agents. Our methods belong to this direction, but differ from prior methods as we create an imitator that uses imitation learning to mimic and predict victim's intention — this prediction can be used to further strengthen the adversarial agent training.

Imitation learning. A core component of our algorithms is an imitation learning model trained to mimic and product similar victim's actions. We employed an extended version of the Generative Adversarial Imitation Learning (GAIL) algorithm [9, 25], a state-of-the-art imitation leaning algorithm that is highly scalable for continuous domains such as the MuJoCo ones. The literature on learning from expert demonstrations covers both imitation learning and inverse reinforcement learning (IRL) works. While imitation learning presents a direct approach to imitate expert's policies, IRL [4–6, 31] assumes that expert's policy is driven by an expert's reward function, thus propose to infer this function from demonstrations. Even-though IRL is more transferable for changing environments, it is often less effective in mimicking expert's demonstrations [9]. On the other hand, for the MuJoCo game environments, since the rewards are obvious, imitation learning (or specifically GAIL) provides us with a direct and suitable algorithm for imitating the victim. Note that, in our settings, the victim's environmental dynamics keep changing during the adversarial training, raising a need for redesigning the GAIL, in both theoretical and practical aspects. We address this issue later in this paper.

3 ADVERSARIAL POLICY FRAMEWORK

Following the general framework introduced in [7], we consider a two-player Markov game in which the victim plays against an opponent which is under control by the adversary. We thus name these two players the victim agent and adversary agent. We represent the two-player non-zero-sum Markov game as a tuple:

$$(\mathcal{S}, \mathcal{A}^{\alpha}, \mathcal{A}^{\nu}, \mathbf{q}^{\alpha}, \mathbf{q}^{\nu}, \mathbf{r}^{\alpha}, \mathbf{r}^{\nu}, \gamma),$$

where α refers to the adversary and ν refers to the victim. In addition, $\mathcal{S}$ is the set of the states, $(\mathcal{A}^{\alpha}, \mathcal{A}^{\nu})$ are sets of actions, $(\mathbf{q}^{\alpha}, \mathbf{q}^{\nu})$ are transition probabilities and $(\mathbf{r}^{\alpha}, \mathbf{r}^{\nu})$ are reward functions of the two players, respectively. Finally, $\gamma \in [0, 1]$ is a discount factor. The objective of each player is to maximize his/her long-term expected reward. Essentially, given a policy of the victim, denoted by π^{ν}, the adversary aims at finding an optimal policy π^{α} that maximizes their long-term reward, formulated as follows:

$$\max_{\pi^{\alpha}} \left\{ V_{\pi^{\alpha}}(s_0 | \mathbf{q}^{\alpha}(\pi^{\nu})) = \mathbb{E}_{\tau \sim \pi^{\alpha}} \left[\sum_{t=0}^{\infty} \gamma^t r^{\alpha}(s_t) \,\middle|\, \mathbf{q}^{\alpha}(\pi^{\nu}) \right] \right\},$$

where s_0 is the initial state and τ denotes a trajectory sampled from executing the adversary policy π^{α} in the environment and $s_t \in \tau$ for all t. Transition probabilities of the adversary, denoted by $\mathbf{q}^{\alpha}(\pi^{\nu})$, depends on the policy of the victim π^{ν}. Specifically, the transition probability $q^{\alpha}(s_{t+1}|s_t, a^{\alpha}_t)$ can be generally computed as follows:

$$q^{\alpha}(s_{t+1}|s_t, a^{\alpha}_t) = \sum_{a^{\nu} \in \mathcal{A}^{\nu}} \pi^{\nu}(a^{\nu}_t|s_t) P(s_{t+1}|s_t, a^{\alpha}_t, a^{\nu}_t),$$

where $P(s_{t+1}|s_t, a^{\alpha}, a^{\nu}_t)$ is the probability of reaching state s_{t+1} if the adversary and victim agents take action a^{α}_t, a^{ν}_t, respectively, at state s_t. In other words, if the policy of the victim is fixed, then the transition probabilities $\mathbf{q}^{\alpha}$ are also fixed. Similarly, the objective of the victim is to maximize the expected long-term rewards of the victim, formulated as follows:

$$\max_{\pi^{\nu}} \left\{ V_{\pi^{\nu}}(s_0 | \mathbf{q}^{\nu}(\pi^{\alpha})) = \mathbb{E}_{\tau \sim \pi^{\nu}} \left[\sum_{t=0}^{\infty} \gamma^t r^{\nu}(s_t) \,\middle|\, \mathbf{q}^{\nu}(\pi^{\alpha}) \right] \right\}.$$

Intuitively, if the policy of one player is fixed, then the transition probabilities (or dynamics) of the other player's environment is also fixed; thus the two-player game becomes a standard RL task.

In this paper, we assume the victim follows a fixed policy (which was pre-trained). This is a common assumption in adversarial policy learning research, motivated by real-world settings such as autonomous vehicles in which RL-trained policies might be deployed [8]. We will later discuss the effect of unfixed victim's policies on the adversarial policy training. As mentioned previously, existing work directly trains adversarial policies based on interactions with the victim. We instead create an imitator who follows imitation learning to discover underlying characteristics of the victim policy and transfers that knowledge to the adversary, helping the adversary produces a better adversarial policy. Both the imitator and the adversary policies will be trained simultaneously based on interactions between the adversary and the victim.

Next, we will first present our imitation learning of the victim policy. We then follow with the elaboration on our adversarial policy learning that incorporates the victim imitation learning component.

4 VICTIM IMITATION LEARNING

In standard imitation learning, we learn to imitate an expert (which is the victim in our study) based on a fixed set of trajectories sampled from the expert's policy. On the other hand, in our problem, learning the victim policy is more challenging since it involves the adversarial policy which is also being trained at the same time (our observations of the victim policy depend on what policy the adversary is playing). In the following, we first introduce our advanced imitation model and algorithm given a fixed adversary policy. We then present our theoretical results on the impact of the adversary policy (during the training process) on our imitation learning.

4.1 Enhanced Imitation Learning Model

Our objective is to build an imitation learning model to imitate the victim's policy through observing victim trajectories. Our model is essentially an enhanced version of the GAIL algorithm [9]. Overall, following GAIL, given an attacker policy π^α, an imitation policy can be learned by solving the following saddle point problem:

$$\max_{\widetilde{\pi}^v_\psi} \min_{D_w} \Big\{ \phi(\widetilde{\pi}^v_\psi, D_w) = \mathbb{E}_{\tau\sim\widetilde{\pi}^v_\psi}\Big[\sum\nolimits_t \log(D_w(s_t,a^v_t)) \,\Big|\mathbf{q}^v(\pi^\alpha)\Big] + \mathbb{E}_{\tau\sim\pi^v}\Big[\sum\nolimits_t \log(1-D_w(s_t,a^v_t))\,\Big|\mathbf{q}^v(\pi^\alpha)\Big] - \lambda H(\widetilde{\pi}^v_\psi)\Big\} \tag{1}$$

where $H(\cdot)$ is the entropy function, $\widetilde{\pi}^v_\psi$ refers to the imitating policy which is an output of a neural net with parameter ψ, and π^v is the victim's policy to be imitated. In addition, D is a discriminative neural net model with parameter w to distinguish between trajectories generated by $\widetilde{\pi}^v_\psi$ and those from the victim's policy. Normally, GAIL would require a large amount of victim's trajectories to provide a good imitation policy. Since we want the imitation model to work with the adversary's policy optimization simultaneously, this is difficult to achieve at early episodes when the set of demonstrations from the victim is limited. Therefore, we propose to robustify GAIL by adding the adversary's value function to the objective:

$$\max_{\widetilde{\pi}^v_\psi} \min_{D_w} \Big\{ \phi^E(\widetilde{\pi}^v_\psi, D_w|\pi^\alpha) = \phi(\widetilde{\pi}^v_\psi, D_w) - V_{\pi^\alpha}(s_0|\mathbf{q}^\alpha(\widetilde{\pi}^v_\psi))\Big\} \tag{2}$$

In this enhanced model (2), the aim to train a policy that both mimics the victim and minimizes the adversary's long-term reward. The value function $V_{\pi^\alpha}(s_0|\mathbf{q}^\alpha(\widetilde{\pi}^v_\psi))$ is the adversary's expected reward, but defined as a function of imitator's policy. The inclusion of the opponent's value function makes (2) not straightforward to handle and would require a redesign of the objective function to make it practical, as stated in [8]. Despite of that, we can provide, in the following, a simple formulation for the gradient of $V_{\pi^\alpha}(s_0|\mathbf{q}^\alpha(\widetilde{\pi}^v_\psi))$, making it convenient to be handled by a standard policy optimization algorithm, e.g., TRPO or PPO [23, 24].

Lemma 4.1. *The gradient of the value function for the adversary $V_{\pi^\alpha}(s_0|\boldsymbol{q}^\alpha(\widetilde{\pi}^v_\psi))$ w.r.t ψ can be computed as follows:*

$$\nabla_\psi\Big(V_{\pi^\alpha}(s_0|\boldsymbol{q}^\alpha(\widetilde{\pi}^v_\psi))\Big) = \mathbb{E}_{\tau\sim\widetilde{\pi}^v_\psi}\Big[R^\alpha(\tau)\sum\nolimits_t \nabla_\psi \log\widetilde{\pi}^v_\psi(a^v_t|s_t)\Big|\boldsymbol{q}^v(\pi^\alpha))\Big],$$

where $R^\alpha(\tau) = \sum_t \gamma^t r^\alpha(s_t)$ with $s_t \in \tau$.

As a result, at each imitation learning step, after updating the discriminator $D_w(s_t, a^v_t)$, one can update the imitating policy $\widetilde{\pi}^v_\psi$ using the gradient given in Proposition (4.2) below.[2]

Proposition 4.2. *The gradient of the objective* (2) *w.r.t. ψ can be computed as follows:*

$$\mathbb{E}_{\tau\sim\widetilde{\pi}^v_\psi}\Big[\sum\nolimits_t \gamma^t\eta(s_t,a^\alpha_t,a^v_t)\sum\nolimits_t\nabla_\psi\log\widetilde{\pi}^v_\psi(a^v_t|s_t)\Big] - \lambda\nabla_\psi H(\widetilde{\pi}^v_\psi)$$

where $\eta(s_t,a^\alpha_t,a^v_t) = \log(D_w(s_t,a^v_t)) - r^\alpha(s_t)$.

With all the findings above, we can show that the enhanced imitation learning model (2) can be converted into a standard GAIL with a modified objective function, with a note that the imitation learning model depends on the adversary's policy π^α, which dictates the dynamics of the victim's environment.

Corollary 4.3. *The enhanced imitation learning model* (2) *is equivalent to GAIL with the modified objective:*

$$\max_{\widetilde{\pi}^v} \min_{D_w} \Big\{ \phi^E(\widetilde{\pi}^v_\psi, D_w|\pi^\alpha) = \mathbb{E}_{\tau\sim\widetilde{\pi}^v_\psi}\Big[\sum\nolimits_t \eta(s_t,a^\alpha_t,a^v_t)\,\Big|\boldsymbol{q}^v(\pi^\alpha)\Big] + \mathbb{E}_{\tau\sim\pi^v}\Big[\sum\nolimits_t \log(1-D_w(s_t,a^v_t))\,\Big|\boldsymbol{q}^v(\pi^\alpha)\Big] - \lambda H(\widetilde{\pi}^v_\psi)\Big\} \tag{3}$$

4.2 Imitation Learning Algorithm

With all the theoretical results on gradient computation developed in the previous section, we are now ready for the victim imitation learning algorithm. As shown in Corollary 4.3, the enhanced imitation learning model can be converted to a standard one, implying that the same optimization steps in [9] can be used with the the modified discriminator's objective. That is, at each iteration of the adversarial policy optimization, one can follow the following three steps to update the imitating policy $\widetilde{\pi}^v_\psi$:

(i) Sample imitation trajectories $\tau^v_i \sim (\widetilde{\pi}^v_\psi, \pi^\alpha)$.

(ii) Update the discriminator $D_w(s, a^v)$ with the gradients:

$$\mathbb{E}_{\tau^v_i}\Big[\sum_t \gamma^t\nabla_w\log(D_w(\cdot))\Big] + \mathbb{E}_{\tau^v_E}\Big[\sum_t\gamma^t\nabla_w\log(1-D_w(\cdot))\Big] \tag{4}$$

where τ^v_E are historical trajectories of the victim collected from interactions between the adversary and the victim.

[2] Detailed proofs of all theoretical results are in the appendix.

(iii) Update ψ with the gradients:

$$\mathbb{E}_{\tau_i^\nu}\left[\sum_t \gamma^t \eta(s_t, a_t^\alpha, a_t^\nu)\sum_t \nabla_\psi \log \pi_\psi^\nu(a_t^\nu|s_t)\right] - \lambda\nabla_\psi H(\widetilde{\pi}_\psi^\nu), \quad (5)$$

which is a standard policy gradient update, for which one can use TRPO [23] or PPO [24].

For all the updates above, we interact with an adversary of policy π^α. This policy will keep changing during the adversarial policy learning and affect the victim's environmental dynamics. This would make the imitation learning process unstable and challenging to handle. We analyze the effect of the adversary's policy on the imitating policy in Section 4.3.

4.3 Effects of the Adversary's Policy on the Victim Imitation Policy

It is important to see that our imitation learning differs from the standard GAIL as the dynamics $\mathbf{q}^\nu(\pi^\alpha)$ are dependent of the adversary policy π^α and our imitation learning model will be trained simultaneously with the adversary's policy. This raises questions of how the learning of the imitating policy is affected by such changing dynamics, and whether one can get a desired imitating policy when the adversary's policy gets stable. To answer these questions, let use consider the following victim's expected reward as a function of adversary's policy.

$$\Gamma(\pi^\alpha) = \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}^\nu(\pi^\alpha)\right].$$

That is, $\Gamma(\pi^\alpha)$ is the expected reward that the victim can get by running a fixed policy π^ν when the adversary policy is π^α. Lemma 4.4 establishes a bound for the gap $|\Gamma(\pi^\alpha) - \Gamma(\widetilde{\pi}^\alpha)|$, which implies that $\Gamma(\widetilde{\pi}^\alpha)$ will converge to $\Gamma(\pi^\alpha)$ if $\widetilde{\pi}^\alpha$ gets close to π^α.

Lemma 4.4. *Given two adversary policies π^α and $\widetilde{\pi}^\alpha$, let $\mathcal{H} = \max_s\{|V_{\pi^\nu}(s|\ \boldsymbol{q}^\nu(\pi^\alpha))|\}$. We obtain the following bound:*

$$\left|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)\right| \leq \frac{\gamma\mathcal{H}\sqrt{2\ln 2}}{1-\gamma}\max_{s\in S}\left\{\sqrt{D_{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}$$

where $D_{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))$ is the KL divergence between the two adversary policies $\widetilde{\pi}^\alpha$ and π^α.

To prove the above lemma, we extend the concept of the advantage function popularly used in single-agent RL [11, 23] to introduce the following victim's *competitive advantage function*, for any two states $s, \bar{s}$, conditional on adversary's policy π^α,

$$A_{\pi^\alpha}(s,\bar{s}) = r^\nu(s) + \gamma V_{\pi^\nu}(\bar{s}|\ \mathbf{q}^\nu(\pi^\alpha)) - V_{\pi^\nu}(s|\ \mathbf{q}^\nu(\pi^\alpha)).$$

This allow use to write the expected reward $\Gamma(\pi^\alpha)$ in terms of the expected long-term competitive advantage function over another adversary policy $\widetilde{\pi}^\alpha$.

$$\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha) = \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_{t=0}\gamma^t\left(A_{\pi^\alpha}(s_t, s_{t+1})\right)\middle|\ \mathbf{q}^\nu(\widetilde{\pi}^\alpha)\right]$$

with a note that $\mathbb{E}_{s_{t+1}\sim\mathbf{q}(\pi^\alpha)}[A_{\pi^\alpha}(s_t, s_{t+1})] = 0$. This identity expresses the expected return of the adversary policy $\widetilde{\pi}^\alpha$ over another policy π^α in terms of victim's expected rewards. The competitive advantage function can be further bounded as

$$\mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}[A_{\pi^\alpha}(s,\bar{s})] \leq \gamma\mathcal{H}\max_s ||\pi^\alpha(\cdot|s) - \widetilde{\pi}^\alpha(\cdot|s)||_1,$$

which can further bounded by $\gamma\mathcal{H}\max_s\left\{\sqrt{2\ln 2D_{\text{KL}}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}$. The full proof is given in the appendix. The proof of Lemma 4.4 also reveals a bound based on maximum norm $|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)| \leq \frac{\mathcal{H}\gamma}{1-\gamma}||\widetilde{\pi}^\alpha - \widetilde{\pi}^\alpha||_\infty$, implying that $\Gamma(\pi^\alpha)$ is Lipschitz continuous in π^α with Lipschitz constant $\frac{\mathcal{H}\gamma}{1-\gamma}$.

Now, let $\pi^{\alpha*}$ be a target adversary's policy that the imitator should be trained with. This would be a trained adversary's policy after the adversarial policy learning. If the imitating policy is trained with another adversary's policy, we aim to explore how this imitating policy performs under the target policy $\pi^{\alpha*}$. Theorem 5.3 below establishes a performance guarantee for the imitating policy if it is trained with a different adversary's policy.

Theorem 4.5. *Suppose that discriminator's network model D of (2) varies within $[D^L, D^U] \subset [0,1]$. Let $\pi^{\alpha*}$ be the target adversary policy that we want to train the imitation policy with, and let $(\widetilde{\pi}^{\nu*}, D^{\nu*})$ be the imitation policy and the imitator's discriminator trained with another adversary π^α, we have the following performance guarantee for $\widetilde{\pi}^{\nu*}$.*

$$\begin{aligned}\left|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right|\\ \leq 2K\max_{s\in S}\left\{\sqrt{D_{KL}(\pi^\alpha(\cdot|s)||\pi^{\alpha*}(\cdot|s))}\right\},\end{aligned}$$

where

$$K = \frac{\gamma\sqrt{2\ln 2}\left(\max_s\{r^\nu(s)\} - \log(D^L - D^LD^U)\right)}{(1-\gamma)^2}.$$

Since the adversary policy π^α will keep changing during our adversarial policy optimization, Theorem 4.5 implies that the imitating policy will be stable if π^α becomes stable, and if π^α is approaching the target adversary's policy, the imitator's policy also converges to the one that is trained with the target adversary policy with rate $O\left(\sqrt{D_{\text{KL}}(\pi^\alpha||\pi^{\alpha*})}\right)$. In other words, if the actual policy π^α is not not too far from the target $\pi^{\alpha*}$ such that $D_{\text{KL}}(\pi^\alpha||\pi^{\alpha*}) \leq \epsilon$, then the expected return of the imitating policy is within a $O(\sqrt{\epsilon})$ neighbourhood of the desired "expected return".

5 ADVERSARIAL POLICY TRAINING

We now discuss our main adversarial policy learning algorithm. We start by explaining the adversarial policy model introduced in [8], upon which we build our new adversarial policy learning algorithm. We then introduce our integration of the victim imitator and our new corresponding main learning algorithm. Finally, we provide our theoretical analysis on the worst-case performance of our learning algorithm when the victim's policy is not fixed.

5.1 Adversarial Policy Learning with Integration of Victim Imitator

Similar to prior works, we assume that the policy of victim is fixed and the aim is to learn an adversary policy to maximize the chances of winning (or win and tie). Similarly to [8], we train the policy by maximizing the following enhanced objective

$$\max_{\pi^\alpha}\left\{V_{\pi^\alpha}(s_0) - V_{\pi^\nu}(s_0|\mathbf{q}^\nu(\pi^\alpha))\right\} \quad (6)$$

where $V_{\pi^\alpha}(s_0)$ is the expected long-term reward of the adversary by following the policy π^α but the transition probabilities are affected by the victim policy π^ν. The objective in (6) involves both the value functions of the adversary and victim, in which the value function of the victim depends on the adversary's policy though the environment dynamics. This complication makes (6) not straightforward to solve. In Proposition 5.1 below we show how to compute the policy gradient of the enhanced adversarial training model in (6), based on which we can show the RL problem in (6) can be converted into a standard competitive game.

Proposition 5.1. *The gradient of* (6) *w.r.t adversary's policy can be computed as follows:*

$$\nabla_\theta \left(V_{\pi^\alpha_\theta}(s_0) - V_{\pi^\nu}(s_0|\boldsymbol{q}^\nu(\pi^\alpha_\theta)) \right)$$

$$= \mathbb{E}_{\tau\sim(\pi^\nu,\pi^\alpha)} \left[\Delta^R(\tau) \sum_t \nabla_\theta \log \pi^\alpha_\theta(a^\alpha_t|s_t) \right].$$

where $\Delta^R(\tau) = \sum_t \gamma^t \left[r^\alpha(s_t) - r^\nu(s_t) \right]$

Similarly to Corollary 4.3, the RL problem in (6) can be converted into a standard competitive game with differentiated rewards $r^\alpha(s_t) - r^\nu(s_t)$ and fixed victim's policy π^ν. Thus, the environmental dynamics are fixed and a standard RL algorithm can apply.

Corollary 5.2. (6) *is equivalent to*

$$\max_{\pi^\alpha} \left\{ \mathbb{E}_{\tau\sim\pi^\alpha} \left[\sum_{t=0}^{\infty} \gamma^t \Delta^r(s_t) \,\middle|\, \boldsymbol{q}^\alpha(\pi^\nu) \right] \right\} \tag{7}$$

where $\Delta^r(s_t) = r^\alpha(s_t) - r^\nu(s_t)$.

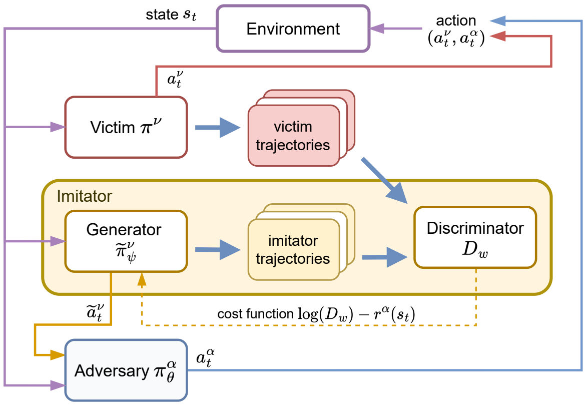
Integrating the victim imitator. To integrate the imitation learning model to the adversarial policy optimization, we use the imitating policy $\widetilde{\pi}^\nu$ to predict victim's intention (i.e., next actions) and include this information into the state space of the adversary. Intuitively, we support the adversary by providing it more information about the victim's next moves. Based on Corollary 5.2, we train the adversary's policy by solving the optimization problem (7) where the adversary's policy is now of the form $\pi^\alpha\left(a^\alpha_t|s_t, \widetilde{a}^\nu_t\right)$. That is, the adversary's policy now is conditional on current state s_t as well as a predicted next victim action $\widetilde{a}^\nu_t$ provided by the imitating policy model $\widetilde{\pi}^\nu$, and the imitating policy model $\widetilde{\pi}^\nu$ takes the state s_t to predict the next victim's actions. Finally, (7) can be solved using standard policy gradient algorithms such as PPO.

Main learning algorithm. Putting all the results developed above together, we present our adversarial policy learning in Algorithm 1. Figure 5 illustrates the three components of our algorithm, including the adversary, the victim and the imitator, and connections between these three components. In short, both the adversary policy and imitator policy are simultaneously updated during interactions between the adversary agent and the victim agent. Observed trajectories are transferred to the imitator to update the imitating policy, following steps in Section 4.2. Simultaneously, our algorithm provides the imitator with the victim's current state to ask for victim's intention. This information will be passed to the adversary's policy network to update the policy adversarial learning. It is expected that when the adversary's policy gets stable and demonstrations from the victim agent are sufficient, the imitation policy also gets close to a desired one (as shown in Section 4.3) and the imitator is able to accurately predict victim agent's next actions. Algorithm 1 shows the main steps of our adversarial policy training algorithm and we give a more detailed version in the appendix.

Algorithm 1 Adversarial Policy Imitation Learning (brief version)

```
Input: Adversary's policy network $\pi^\alpha_\theta$; imitator's policy network $\widetilde{\pi}^\nu_\psi$; imitator's discriminator $D_w$; initial parameters $\theta_0, \psi_0, w_0$.
for $i = 0, 1, 2, \ldots$ do
    # Updating imitator's policy
    Sample trajectories $\tau_i \sim \pi^\alpha_{\theta_i}, \pi^\nu$.
    Update discriminator $D_w$ network from $w_i$ to $w_{i+1}$ using (4).
    Update imitator's policy network from $\psi_i$ to $\psi_{i+1}$ based on TRPO or PPO using (5).
    # Updating adversary's policy
    Generate imitator's predicted actions $\widetilde{a}^\nu \sim \pi^\alpha_{\psi_{i+1}}$
    Update the adversary policy from $\theta_i$ to $\theta_{i+1}$ based on TRPO or PPO, using the gradients in (5.1).
end for
```

Figure 1: An overview of our adversary policy training algorithm. At each step t, state s_t is forwarded to victim policy π^ν and to the imitator's generator policy $\widetilde{\pi}^\nu_\psi$ to generate the victim action a^ν_t and the imitator action $\widetilde{a}^\nu_t$ respectively. The adversary policy π^α_θ uses this imitator output $\widetilde{a}^\nu_t$ and current state s_t to generate the adversary action a^α_t. The environment then transits to the next state given the combination of actions (a^ν_t, a^α_t). For each step, victim and imitator transitions are appended to corresponding trajectory buffers, which will be used to update the discriminator of the imitator D_w.



5.2 Worst-case Performance When Training the Adversary with Unfixed Victim's Policy

So far we train the adversary policy by assuming that the victim agent always follows a fixed policy. We explore, in this section, the question that, if the victim's policy is not fixed, how the victim's unstable policy would affect the adversarial policy learning. To start our analysis, let π^ν_0 be a "true" victim policy that the adversary agent should be trained with and suppose that, due to external causes, the adversary agent is only trained with victim policies that vary within the following set:

$$\Omega(\epsilon) = \{\pi^{\nu} \Big| \max_{s \in \mathcal{S}} D_{\mathrm{KL}}(\pi^{\nu}(s) || \pi_0^{\nu}(s)) \leq \epsilon\}.$$

We define the worst-case expected return of the adversary agent when being trained with such varying victim policies $Y(\epsilon) = \min_{\pi^{\nu} \in \Omega(\epsilon)} \ \max_{\pi^{\alpha}} \left\{\mathbb{E}_{\tau \sim (\pi^{\alpha})} \left[\sum_{t=0}^{\infty} \gamma^t \Delta^r(s_t) \Big| \mathbf{q}(\pi^{\nu})\right]\right\}$. The following theorem gives a bound for the gap between the worst-case and the desired expected return obtained from training with the "true" victim policy $Y^* = \max_{\pi^{\alpha}} \left\{\mathbb{E}_{\tau}\left[\sum_t \gamma^t \Delta^r(s_t) \Big| \mathbf{q}^{\alpha}(\pi_0^{\nu})\right]\right\}$.

Theorem 5.3. *For any $\epsilon > 0$, we have the following bound*

$$\left|Y(\epsilon) - Y^*\right| \leq \frac{\gamma \sqrt{2 \ln 2} \max_s \{|\Delta^r(s)|\}}{(1-\gamma)^2} \sqrt{\epsilon}.$$

The above bound implies that the worst-case performance of the adversarial training would not too bad (i.e., within a neighbourhood $O(\sqrt{\epsilon})$) if the victim policy that the adversary is trained with is not too far from the “true” victim policy. On the other hand, if the adversary is trained with an arbitrary victim policy, the training outcomes would be very bad. Let us use the Rock-paper-scissors game to illustrate this. If the victim always plays “rock” during the adversary’s training, it will not take long for the adversary agent to see that playing “paper” always gives a 100% winning rate. But if the victim change their policy to playing “scissors”, then that trained adversary’s policy will always yield a 0% winning rate.

6 EVALUATION

We evaluate our proposed Enhanced Adversarial Policy Imitation Learning (**E-APIL**) algorithm, i.e., the adversarial policy learning (7) with an enhanced imitator (2), and the non-enhanced imitation version of our algorithm (named, **APIL**), i.e., the adversarial policy learning (7) with a non-enhanced imitator (1). As to do so, we use four competitive MuJoCo game environments introduced by Emergent Complexity (EC) [1], including *Kick And Defend, You Shall Not Pass*, *Sumo Humans*, and *Sumo Ants*. We compare the performance of our algorithms with: (i) the well-trained adversary/victim agents in [1] which we consider as **Baseline** agents; (ii) Attacking Deep Reinforcement Learning (**ADRL**) [7]; and (iii) Adversarial Policy Learning (**APL**) [8]. The two methods, ADRL and APL, are the state-of-the-art methods in adversarial policy learning. For fair comparisons, we use the same experiment settings (i.e., pre-trained parameters, hyperparameters, and evaluation metrics) as in [7, 8]. Implementation details are specified in supplementary section.

6.1 Adversarial Policy Performance: Training Adversary against Baseline Victim

In this experiment, we train our adversary agent using our proposed algorithms to play against the baseline victim agent [1]. We aim to examine if our generated adversarial policy can trigger the victim agent to perform poorly. Table 1 shows the winning rate (i.e., numbers in white cells) and the winning plus tie rate (numbers in gray cells) of our trained adversary agents playing against the *baseline* victim agent, compared to those trained by other adversarial policy algorithms (also against the *baseline* victim). Each reported value is calculated based on 1000 different rounds of game playing.

Figure 2: Illustrative snapshots of a victim (in blue) against normal and adversarial opponents (in red) in SumoHumans simulator. Two players of the baseline method try to get close to each other and butt their opponents to win. However, APL learns to kneel to stay in the ring and its victims may find it harder to knock it down. Our algorithm even learns to stand better with two knees and dodge attacks from the victim.

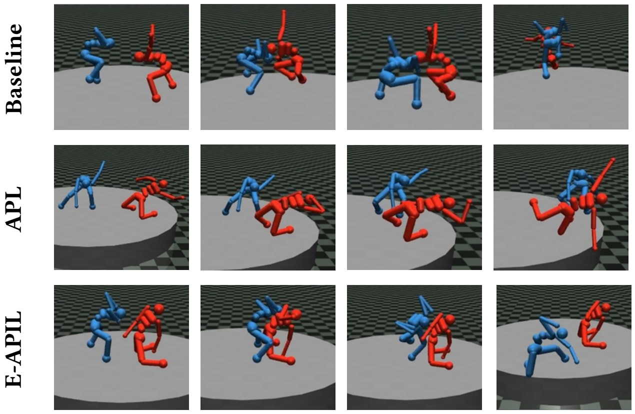
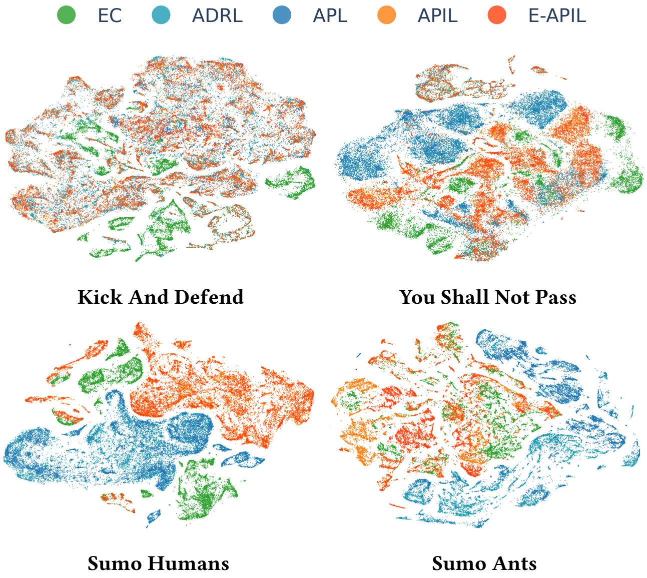


Table 1: Winning rate (white) and winning plus tie rate (gray) of new adversary vs baseline victim.

	Baseline	ADRL	APL	APIL (ours)	E-APIL (ours)
Kick And Defend	28%	48%	80%	85%	**89%**
	29%	49%	80%	86%	**90%**
You Shall Not Pass	38%	56%	**68%**	58%	67%
	38%	56%	**68%**	58%	67%
Sumo Humans	7%	22%	36%	**73%**	72%
	7%	61%	77%	**88%**	87%
Sumo Ants	**39%**	10%	2%	2%	3%
	56%	41%	**81%**	**81%**	80%

Overall, our APIL and E-APIL methods achieve significantly higher winning rates for Kick-And-Defend and Sumo-Humans compared to all existing methods. In You-Shall-Not-Pass, our method E-APIL achieves a winning rate which is only 1% less than the best-performed method (APL) in this game environment, while significantly outperforming the others. On the other hand, in Sumo-Ants, we observe an interesting phenomenon. While we obtain the best winning-plus-tie rates in Sumo-Ants, we obtain a lower winning rate compared to the baseline adversary. This phenomenon also holds true for existing algorithms (ADRL and APL). The cause of this phenomenon comes from a unique underlying characteristic of Sumo-Ants, i.e., it is very challenging to reach the *win* outcome — the victim has a high chance to reach a draw outcome by just jumping to the ground without touching opponent. As a result, our adversary was essentially trained to optimize the policy towards *draw* outcomes in Sumo-Ants, at the sacrifice of the win rate.

Figure 3: t-SNE visualizations of the victim activations when playing against different opponents in MuJoCo games.



Kick And Defend

You Shall Not Pass

Sumo Humans

Sumo Ants

We now seek to better understand why our methods get higher winning rates than other algorithms. Figure 3 shows t-SNE visualization [29] of the trained adversary against the baseline victim by recording victim's policy activations. The t-SNE visualizations for all four game environments indicates that our algorithms APIL/E-APIL seek to activate different policy distribution regions of the victim (the orange and red regions) compared to existing algorithms, allowing our policy learning to converge to a better optimum.

Table 1 also shows that ADRL and APL are more focused on getting draw in the Sumo games than learning how to win. We plot in Figure 4 the training performance of our algorithms for the four game environments, which show that the tie rates are already high during early episodes. As mentioned, the victim in these games can easily get a draw by just jumping to the ground without touching the opponent, which makes the tie rates very high.

Finally, Table 1 shows that our E-APIL with an enhanced imitator is significant better than APIL in both Kick-and-Defend and You-Shall-Not-Pass. This result implies that incorporating the adversary's expected rewards into the imitator's value function definitely helps improve the quality of the generated adversarial policy. In Sumo-Humans and Sumo-Ants where the tie rates account for a large proportion of the outcomes, the performance of E-APIL and APIL are not substantially disparate.

6.2 Blinding the Trained Adversary

To further understand the role of the imitator behind the efficiency of our adversarial training algorithms, we conduct the following experiment. First, we take the trained adversary agents and let them play with the baseline victim, but now we blind the adversary's observation on the victim or, in other words, zero out the adversary observation pertaining to the victim. By blinding the adversary agents, we aim to demonstrate that the trained adversary still manages to make the victim to perform poorly just based on the imitator's policy output, despite of the *blinded* disadvantage.

Table 2: Winning rate (white) and winning plus tie rate (gray) of games between blinded adversary and baseline victim

	Base-line	ADRL	APL	APIL (ours)	E-APIL (ours)
Kick And Defend	28%	48%	80%	85%	**89%**
	29%	49%	80%	86%	**90%**
You Shall Not Pass	1%	48%	65%	62%	**68%**
	1%	48%	65%	62%	**68%**
Sumo Humans	3%	0%	1%	**69%**	66%
	7%	**83%**	62%	**83%**	80%
Sumo Ants	**13%**	7%	3%	2%	3%
	42%	38%	64%	**81%**	79%

Tables 2 reports our experiments with blinded trained adversary against the baseline victim. In general, our APIL and E-APIL adversary agents outperform those trained by the other methods when playing against the baseline victim. Intuitively, even when being blinded, by taking feedback from the trained imitator, our trained adversary agents would still partially predict victim's intention to make better actions, compared to those trained by other methods.

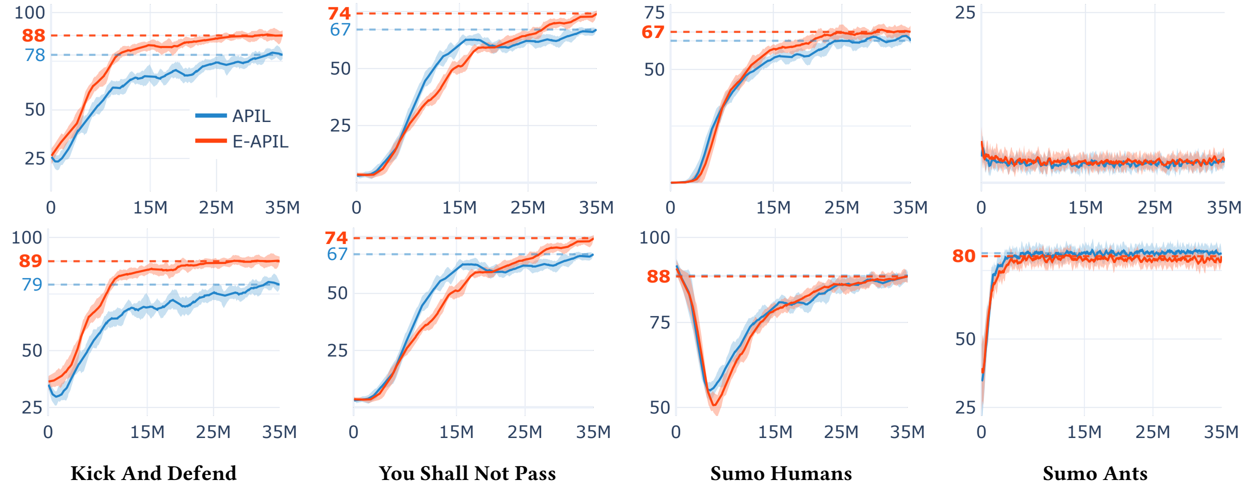
In summary, our methods works well in interactive environments, even in the blinding setting, thanks to the capability of predicting opponent's intention through the trained imitator.

6.3 Improving Victim Resiliency: Retraining Victim against New Adversary

Previous studies demonstrate that one could retrain the victim and thus improve its adversary resistance [7, 8]. In this experiment, we also retrain the victim agent against the newly trained adversary agent to examine the resistance of the retrained victim agent against adversarial policies. We further explore the resilience transferability of the retrained victim agents. Specifically, similar to previous work, we retrain the victim agent against a mixed adversary agent of the new adversary (whose policy is trained based on one of the evaluated adversarial training algorithms (i.e., baseline, ADRL, APL and ours) and the baseline adversary. We then have the retrained victim agent play against the baseline adversary for 1000 rounds and report its winning as well as winning plus tie rates.

Table 3 shows the winning and winning plus tie rates of the retrained victim agents playing against the baseline adversary. For example, the **E-APIL** column shows the game results between the victim agent (retrained based on interactions with an E-APIL adversary) and the Baseline adversary agent. Except for the Sumo-Ants game, our methods outperform other algorithms in terms of retraining the victim agent to be stronger. For all the games, the baseline victim generally yields good results, which is not a surprising observation as the OpenAI's baseline agents [1] are well trained against their opponents with around 1B steps and 4 GPUs. In our experiments, with only 35M + 10M steps and 1 GPU, we are able to make the victim agent significantly stronger. Moreover, despite the fact that the APIL/E-APIL based victim agents are retrained against our APIL/E-APIL adversary agents, these retrained victim agents still manage to perform well against the baseline adversary

Figure 4: Performance of newly trained adversary vs. baseline victim while training against baseline victim. First row: win-rate. Second row: win-rate + tie-rate. Blue curves: AIPL, red curves: E-AIPL. In *You-Shall-Not-Pass-Humans*, the tie rates are always zero because there is no declaration for a tie game.



as show in the last two columns of Table 3. This result clearly shows that the strong resilience of APIL/E-APIL based victim agents can be transferred to other game settings with different types of adversary agents (e.g., the baseline adversary in this experiment).

Table 3: Winning rate (white) and winning plus tie rate (gray) of retrained victim agents vs baseline adversary.

	Base-line	ADRL	APL	APIL (ours)	E-APIL (ours)
Kick And Defend	71%	62%	70%	**87%**	**87%**
	72%	70%	77%	89%	**90%**
You Shall Not Pass	62%	64%	63%	71%	**72%**
	62%	64%	63%	71%	**72%**
Sumo Humans	93%	76%	79%	94%	**95%**
	93%	85%	84%	95%	**96%**
Sumo Ants	**44%**	24%	30%	29%	33%
	61%	38%	48%	52%	55%

We further test the performance of each retrained victim agent against our E-APIL adversary and report the winning and winning plus tie rates in Table 4. For the two non-sumo games, the winning rates of ADRL/APL retrained victims are less than 60%, which are significantly smaller than our rates. Obviously, our E-APIL retrained victim achieves better results because it's trained against our E-APIL adversary, but our APIL also gets much better winning rates than ADRL/APL methods. It generally indicates the robustness and efficiency of our algorithms, compared to other approaches, in terms of retraining the victim agent to have better versions of it.

7 CONCLUSION

This paper introduces a new effective adversarial policy learning algorithm based on a novel integration of a new victim-imitation learning into the adversarial policy training process. Our victim-imitation component (which is an enhanced version of the state-of-the-art imitation method GAIL) discovers underlying characteristics of the victim agent, enabling the prediction of the victim's next moves which can be leveraged to strengthen the adversarial policy generation. We present important theoretical results on the inter-dependency between the victim-imitation learning and the adversarial policy learning, showing the convergence of our learning algorithm. We demonstrate the superiority of our proposed algorithm compared to existing adversarial policy learning algorithms through extensive experiments on various game environments.

Table 4: Winning rate (white) and winning plus tie rate (gray) of retrained victim agents vs our E-APIL adversary

	Base-line	ADRL	APL	APIL (ours)	E-APIL (ours)
Kick And Defend	10%	31%	52%	82%	**91%**
	11%	34%	53%	83%	**92%**
You Shall Not Pass	33%	46%	58%	78%	**88%**
	33%	46%	58%	78%	**88%**
Sumo Humans	14%	32%	46%	**51%**	41%
	28%	84%	85%	84%	**91%**
Sumo Ants	**20%**	17%	19%	**20%**	**20%**
	97%	96%	**97%**	**97%**	**97%**

8 ACKNOWLEDGMENT

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-017). Dr. Nguyen was supported by grant W911NF-20-1-0344 from the US Army Research Office.

REFERENCES

[1] Trapit Bansal, Jakub W. Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. 2018. Emergent Complexity via Multi-Agent Competition. *ArXiv* abs/1710.03748 (2018).

[2] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 262–275.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[4] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852* (2016).

[5] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*. PMLR, 49–58.

[6] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248* (2017).

[7] Adam Gleave, Michael Dennis, Neel Kant, Cody Wild, Sergey Levine, and Stuart J. Russell. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. *ArXiv* abs/1905.10615 (2020).

[8] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. 2021. Adversarial Policy Learning in Two-player Competitive Games. In *ICML*.

[9] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

[10] Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and P. Abbeel. 2017. Adversarial Attacks on Neural Network Policies. *ArXiv* abs/1702.02284 (2017).

[11] Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer.

[12] Jernej Kos and Dawn Song. 2017. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452* (2017).

[13] Xian Yeow Lee, Sambit Ghadai, Kai Liang Tan, Chinmay Hegde, and Soumik Sarkar. 2020. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4577–4584.

[14] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125* (2017).

[15] Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. 2020. On the robustness of cooperative multi-agent reinforcement learning. In *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 62–68.

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

[17] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. 2019. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems* 32 (2019).

[18] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. 2018. Reinforcement learning for solving the vehicle routing problem. *Advances in neural information processing systems* 31 (2018).

[19] Laura Noonan. 2017. JPMorgan develops robot to execute trades. *Financial Times* (2017), 1928–1937.

[20] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. 2020. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*. PMLR, 7974–7984.

[21] Amin Rakhsha, Xuezhou Zhang, Xiaojin Zhu, and Adish Singla. 2021. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *arXiv preprint arXiv:2102.08492* (2021).

[22] Alessio Russo and Alexandre Proutiere. 2019. Optimal attacks on reinforcement learning policies. *arXiv preprint arXiv:1907.13548* (2019).

[23] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[25] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems* 31 (2018).

[26] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. 2020. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5883–5891.

[27] MTCAJ Thomas and A Thomas Joy. 2006. *Elements of information theory*. Wiley-Interscience.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 5026–5033.

[29] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[30] Chaowei Xiao, Xinlei Pan, Warren He, Jian Peng, Mingjie Sun, Jinfeng Yi, Mingyan Liu, Bo Li, and Dawn Song. 2019. Characterizing attacks on deep reinforcement learning. *arXiv preprint arXiv:1907.09470* (2019).

[31] Lantao Yu, Jiaming Song, and Stefano Ermon. 2019. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*. PMLR, 7194–7201.

[32] Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. 2021. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452* (2021).

[33] Yiren Zhao, Ilia Shumailov, Han Cui, Xitong Gao, Robert Mullins, and Ross Anderson. 2020. Blackbox attacks on reinforcement learning agents using approximated temporal information. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 16–24.

Appendix

A MISSING PROOFS

A.1 Proof of Lemma 4.1

Lemma A.1. *The gradient of $V_{\pi^\alpha}(s_0|\boldsymbol{q}^\alpha(\widetilde{\pi}^\nu_\psi))$ w.r.t ψ can be computed as follows:*

$$\nabla_\psi \left(V_{\pi^\alpha}(s_0|\boldsymbol{q}^\alpha(\widetilde{\pi}^\nu_\psi)) \right) = \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu_\psi} \left[R^\alpha(\tau) \sum_t \nabla_\psi \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \middle| \boldsymbol{q}^\nu(\pi^\alpha) \right]. \tag{8}$$

Proof. We write the adversary's expected reward as

$$\begin{aligned} \mathbb{E}_{\tau\sim\pi^\alpha} \left[\sum_t \gamma^t r^\alpha(s_t) \middle| \mathbf{q}^\alpha(\widetilde{\pi}^\nu_\psi) \right] &= \sum_\tau R^\alpha(\tau) \prod_t \pi^\alpha(a^\alpha_t|s_t) P(s_{t+1}|s_t, a^\alpha) \\ &= \sum_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}} R^\alpha(\tau) \prod_t \pi^\alpha(a^\alpha_t|s_t) \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) P(s_{t+1}|s_t, a^\alpha_t, a^\nu_t) \end{aligned} \tag{9}$$

Taking the derivative of the above expected value w.r.t. ψ we get

$$\begin{aligned} \nabla_\psi \left(\mathbb{E}_{\tau\sim\pi^\alpha} \left[\sum_t \gamma^t r^\alpha(s_t) \middle| \mathbf{q}^\alpha(\widetilde{\pi}^\nu_\psi) \right] \right) &= \sum_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}} R^\alpha(\tau) P(\tau) \nabla_\psi \log \left(\prod_t \pi^\alpha(a^\alpha_t|s_t) \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) P(s_{t+1}|s_t, a^\alpha_t, a^\nu_t) \right) \\ &= \sum_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}} R^\alpha(\tau) P(\tau) \sum_t \nabla_\psi \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \\ &= \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu_\psi} \left[R^\alpha(\tau) \sum_t \nabla_\psi \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \middle| \mathbf{q}^\nu(\pi^\alpha) \right], \end{aligned}$$

which is the desired equality. □

A.2 Proof of Proposition 4.2

Proposition A.2. *The gradient of the objective* (2) *w.r.t. ψ can be computed as follows:*

$$\mathbb{E}_{\tau\sim\widetilde{\pi}^\nu_\psi} \left[\sum_t \gamma^t \eta(s_t, a^\alpha_t, a^\nu_t) \sum_t \nabla_\psi \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \right] - \lambda \nabla_\psi H(\widetilde{\pi}^\nu_\psi)$$

where $\eta(s_t, a^\alpha_t, a^\nu_t) = \log(D(s_t, a^\nu_t)) - r^\alpha(s_t)$.

Proof. The first part of (2) is a standard long-term reward whose gradients can be computed as

$$\nabla_\psi \left(\phi(\widetilde{\pi}^\nu_\psi, D) \right) = \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu_\psi} \left[\sum_t \gamma^t \log(D(s_t, a^\alpha_t)) \sum_t \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \middle| \mathbf{q}^\nu(\pi^\alpha) \right]$$

Combine this with the derivation in Lemma 4.1 we get

$$\nabla_\psi \left(\phi^E(\widetilde{\pi}_\psi, D) \right) = \mathbb{E}_{\tau\sim\widetilde{\pi}_\psi} \left[\sum_t \gamma^t \left(\log(D(s_t, a^\alpha_t)) - r^\alpha(s_t) \right) \sum_t \log \widetilde{\pi}^\nu_\psi(a^\nu_t|s_t) \middle| \mathbf{q}^\nu(\pi^\alpha) \right],$$

as desired. □

A.3 Proof of Corollary 4.3

Corollary A.3. *The enhanced imitation learning model* (2) *is equivalent to GAIL with the modified discriminator objective:*

$$\max_{\widetilde{\pi}^\nu} \min_{D\in[0,1]} \left\{ \phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha) = \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu} \left[\sum_t \eta(s_t, a^\alpha_t, a^\nu_t) \, \middle| \boldsymbol{q}^\nu(\pi^\alpha) \right] + \mathbb{E}_{\tau\sim\pi^\nu} \left[\sum_t \log(1 - D(s_t, a^\nu_t)) \, \middle| \boldsymbol{q}^\nu(\pi^\alpha) \right] - \lambda H(\widetilde{\pi}^\nu) \right\} \tag{10}$$

Proof. The corollary can be deduced from Proposition 4.2, or one can write

$$\begin{aligned}
V_{\pi^\alpha}(s_0|\mathbf{q}^\alpha(\widetilde{\pi}^\nu_\psi)) = \mathbb{E}_{\tau\sim\pi^\alpha}\left[\sum_t \gamma^t r^\alpha(s_t)\middle|\mathbf{q}^\alpha(\widetilde{\pi}^\nu_\psi)\right] &= \sum_\tau R^\alpha(\tau)\prod_t \pi^\alpha(a^\alpha_t|s_t)P(s_{t+1}|s_t, a^\alpha) \\
&= \sum_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}} R^\alpha(\tau)\prod_t \pi^\alpha(a^\alpha_t|s_t)\widetilde{\pi}^\nu_\psi(a^\nu_t|s_t)P(s_{t+1}|s_t, a^\alpha_t, a^\nu_t) \\
&= \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu}\left[\sum_t \gamma^t r^\alpha(s_t)\middle|\mathbf{q}^\nu(\pi^\alpha)\right],
\end{aligned}$$

which directly leads the desired equivalence. □

A.4 Proof of Lemma 4.4

Lemma A.4. *Given two adversary policies π^α and $\widetilde{\pi}^\alpha$, let $\mathcal{H} = \max_s \{|V_{\pi^\nu}(s|\ \boldsymbol{q}^\nu(\pi^\alpha))|\}$*

$$\left|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)\right| \le \frac{\gamma\mathcal{H}\sqrt{2\ln 2}}{1-\gamma}\max_{s\in S}\left\{\sqrt{D_{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}$$

where $D_{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))$ is the KL divergence between $\widetilde{\pi}^\alpha$ and π^α.

Proof. Recall that we define $\Gamma(\pi^\alpha)$ as the expected reward of the victim with policy π^ν when the adversary follows policy π^α

$$\Gamma(\pi^\alpha) = \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}^\nu(\pi^\alpha)\right].$$

Given two adversary policies π^α and $\widetilde{\pi}^\alpha$, we define the following victim's *competitive advantage function* $A_{\pi^\alpha}(s,\bar{s})$ for two states $s, \bar{s} \in \mathcal{S}$

$$A_{\pi^\alpha}(s,\bar{s}) = r^\nu(s) + \gamma V_{\pi^\nu}(\bar{s}|\ \mathbf{q}^\nu(\pi^\alpha)) - V_{\pi^\nu}(s|\ \mathbf{q}^\nu(\pi^\alpha)),$$

where $V_{\pi^\nu}(s|\ \mathbf{q}^\nu(\pi^\alpha)) = \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}(\pi^\alpha), s_0 = s\right]$. We then compute the advantage of the adversary policy π^α over $\widetilde{\pi}^\alpha$ but in terms of victim's expected rewards as follows

$$\begin{aligned}
\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha) &= \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}^\nu(\widetilde{\pi}^\alpha)\right] - V_{\pi^\nu}(s_0|\ \mathbf{q}^\nu(\pi^\alpha)) \\
&\overset{(a)}{=} \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[\sum_t \gamma^t r^\nu(s_t)\right] + \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[\sum_t \gamma^t(\gamma V_{\pi^\nu}(s_{t+1}|\ \mathbf{q}^\nu(\pi^\alpha)) - V_{\pi^\nu}(s_t|\ \mathbf{q}^\nu(\pi^\alpha)))\right] \\
&= \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[\sum_t \gamma^t\left(r^\nu(s_t) + \gamma V_{\pi^\nu}(s_{t+1}|\ \mathbf{q}^\nu(\pi^\alpha)) - V_{\pi^\nu}(s_t|\ \mathbf{q}^\nu(\pi^\alpha))\right)\right] \\
&\overset{(b)}{=} \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[\sum_t \gamma^t\left(A_{\pi^\alpha}(s_t, s_{t+1})\right)\right],
\end{aligned} \tag{11}$$

where (a) is due to the fact that

$$\sum_t \gamma^t(\gamma V_{\pi^\nu}(s_{t+1}|\ \mathbf{q}^\nu(\pi^\alpha)) - V_{\pi^\nu}(s_t|\ \mathbf{q}^\nu(\pi^\alpha))) = V_{\pi^\nu}(s_0|\ \mathbf{q}^\nu(\pi^\alpha)),$$

and (b) is due to the definition of the competitive advantage function $A_{\pi^\alpha}(s_t, s_{t+1})$. Here we note that

$$\mathbb{E}_{\bar{s}^\nu\sim\pi^\nu,\mathbf{q}(\pi^\alpha)|s^\nu}\left[A_{\pi^\alpha}(s^\nu,\bar{s}^\nu)\right] = \mathbb{E}_{\bar{s}^\nu\sim\pi^\nu,\mathbf{q}(\pi^\alpha)|s^\nu}\left[r(s^\nu) + \gamma V_{\pi^\nu}(s^\nu|\ \mathbf{q}(\pi^\alpha)) - V_{\pi^\nu}(\bar{s}^\nu|\ \mathbf{q}(\pi^\alpha))\right] = 0.$$

We further have the following bound for the competitive advantage function.

$$\begin{aligned}
\mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)|s}\left[A_{\pi^\alpha}(s,\bar{s})\right] &= \mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[r^\nu(s) + \gamma V_{\pi^\nu}(\bar{s}|\ \mathbf{q}^\nu(\pi^\alpha))\right] - \mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}^\nu(\pi^\alpha)}\left[r^\nu(s) + \gamma V_{\pi^\nu}(\bar{s}|\ \mathbf{q}^\nu(\pi^\alpha))\right] \\
&= \gamma\left(\sum_{\bar{s}} V_{\pi^\nu}(\bar{s}|\ \mathbf{q}^\nu(\pi^\alpha))\left(P(\bar{s}|s,\pi^\nu,\widetilde{\pi}^\alpha) - P(\bar{s}|s,\pi^\nu,\pi^\alpha)\right)\right) \\
&\le \gamma\mathcal{H}\mathbb{E}_{\bar{s}\sim\pi^\nu|s}\left[||\pi^\alpha(\cdot|\bar{s}) - \pi^\alpha(\cdot|\bar{s})||_1\right] \\
&\overset{(c)}{\le} \gamma\mathcal{H}\max_{s\in\mathcal{S}}\left\{\sqrt{2\ln 2\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\},
\end{aligned} \tag{12}$$

where $\mathcal{H} = \max_s\{|V_{\pi^\nu}(s|\mathbf{q}^\nu(\pi^\alpha))|\}$ and (c) is due to the inequality $||p - q||_1 \leq \sqrt{2\ln 2 D_{\mathrm{KL}}(p||q)}$ for two distributions p, q [27]. Moreover, if we define

$$\epsilon = \max_s \left\{\left|\mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}(\widetilde{\pi}^\alpha)|s}\left[A_{\pi^\alpha}(s,\bar{s})\right]\right|\right\},$$

then $\epsilon \longrightarrow 0$ if $\widetilde{\pi}^\alpha \longrightarrow \pi^\alpha$. Moreover, we can see from (11) that

$$\left|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)\right| = \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}^\nu(\widetilde{\pi}^\alpha)}\left[\sum_t \gamma^t \left(A_{\pi^\alpha}(s_t, s_{t+1})\right)\right] \leq \mathbb{E}_{\tau\sim\pi^\nu,\mathbf{q}(\widetilde{\pi}^\alpha)}\left[\sum_{t=0}^{\infty} \gamma^t \epsilon\right] = \frac{\epsilon}{1-\gamma}. \tag{13}$$

Putting (12) and (13) together, we can bound the gap $|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)|$ as

$$\begin{aligned}\left|\Gamma(\widetilde{\pi}^\alpha) - \Gamma(\pi^\alpha)\right| &\leq \frac{\max_s \left\{\left|\mathbb{E}_{\bar{s}\sim\pi^\nu,\mathbf{q}(\widetilde{\pi}^\alpha)|s}\left[A_{\pi^\alpha}(s,\bar{s})\right]\right|\right\}}{1-\gamma} \\ &\leq \frac{\gamma\mathcal{H}}{1-\gamma} \max_{s\in\mathcal{S}} \left\{\sqrt{2\ln 2\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\},\end{aligned}$$

as desired. □

A.5 Proof of Theorem 4.5

Theorem A.5. *Suppose that discriminator's network model D of* (2) *varies within $[D^L, D^U] \subset [0,1]$. Let $\pi^{\alpha*}$ be the target adversary policy that we want to train the imitation policy with, and let $(\widetilde{\pi}^{\nu*}, D^{\nu*})$ be the imitation policy and the imitator's discriminator trained with another adversary π^α, we have the following performance guarantee for $\widetilde{\pi}^{\nu*}$.*

$$\left|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right| \leq 2K \max_{s\in S}\left\{\sqrt{D_{KL}(\pi^\alpha(\cdot|s)||\pi^{\alpha*}(\cdot|s))}\right\}, \tag{14}$$

where

$$K = \frac{\gamma\sqrt{2\ln 2}\left(\max_s\{r^\nu(s)\} - \log(D^L - D^L D^U)\right)}{(1-\gamma)^2}.$$

Proof. From the proof of Lemma 4.4 we can deduce the following, for any policies π^α, $\widetilde{\pi}^\alpha$, and any reward function $r^\nu(a)$,

$$\begin{aligned}&\left|\mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}^\nu(\pi^\alpha)\right] - \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t r^\nu(s_t)\middle|\mathbf{q}^\nu(\widetilde{\pi}^\alpha)\right]\right| \\ &\quad\leq \frac{\gamma}{1-\gamma}\max_s\{|V_{\pi^\nu}(s|\mathbf{q}^\nu(\pi^\alpha))|\}\max_{s\in\mathcal{S}}\left\{\sqrt{2\ln 2\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\} \\ &\quad\leq \frac{\gamma\max_s|r^\nu(s)|}{(1-\gamma)^2}\left\{\sqrt{2\ln 2\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}.\end{aligned} \tag{15}$$

We now using this to bound the gap $\left|\phi^E(\widetilde{\pi}^\nu, D|\widetilde{\pi}^\alpha) - \phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\right|$ as follows. We first write

$$\begin{aligned}\left|\phi^E(\widetilde{\pi}^\nu, D|\widetilde{\pi}^\alpha) - \phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\right| &\leq \left|\mathbb{E}_{\tau\sim\widetilde{\pi}^\nu}\left[\sum_t \gamma^t(\log(D) - r(s_t))\left|\mathbf{q}^\nu(\widetilde{\pi}^\alpha)\right.\right] - \mathbb{E}_{\tau\sim\widetilde{\pi}^\nu}\left[\sum_t \gamma^t(\log(D) - r(s_t))\left|\mathbf{q}^\nu(\pi^\alpha)\right.\right]\right| + \\ &\quad+\left|\mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t\log(1-D)\left|\mathbf{q}^\nu(\widetilde{\pi}^\alpha)\right.\right] - \mathbb{E}_{\tau\sim\pi^\nu}\left[\sum_t \gamma^t\log(1-D)\left|\mathbf{q}^\nu(\pi^\alpha)\right.\right]\right| \\ &\leq \frac{\gamma\sqrt{2\ln 2}}{(1-\gamma)^2}\max_s\left\{\sqrt{\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}\left(\max_{s,D}\{|r^\alpha(s) - \log(D)\} + \max_D|\log(1-D)|\right) \\ &\overset{(d)}{\leq} \frac{\gamma\sqrt{2\ln 2}}{(1-\gamma)^2}\max_s\left\{\sqrt{\mathrm{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}\left(\max_s\{r(s^\alpha)\} - \log(D^L) - \log(1-D^U)\right),\end{aligned} \tag{16}$$

where (d) is because $D \in [D^L, D^U]$. For ease of notation, let

$$K = \frac{\gamma\sqrt{2\ln 2}\left(\max_{s^\alpha}\{r(s^\alpha)\} - \log(D^L) - \log(1-D^U)\right)}{(1-\gamma)^2};\ \epsilon = \sqrt{D_{\mathrm{KL}}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}.$$

We first try to bound $\left|\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right|$ as follows.

- If $\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} \geq \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}$, then we let $D^* = \text{argmin}_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}$ to have

$$\begin{aligned}\left|\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right| &= \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\} \\ &\leq \phi^E(\widetilde{\pi}^\nu, D^*|\pi^\alpha) - \phi^E(\widetilde{\pi}^\nu, D^*|\pi^{\alpha*}) \\ &\leq K\epsilon \end{aligned} \tag{17}$$

- If $\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} \leq \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}$, then we let $D^* = \text{argmin}_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\}$ to have a similar evaluation

$$\begin{aligned}\left|\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right| &= \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} \\ &\leq \phi^E(\widetilde{\pi}^\nu, D^*|\pi^{\alpha*}) - \phi^E(\widetilde{\pi}^\nu, D^*|\pi^\alpha) \\ &\leq K\epsilon. \end{aligned} \tag{18}$$

So we always have

$$\left|\min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\}\right| \leq K\epsilon. \tag{19}$$

Now, suppose $\pi^{\alpha*}$ is a target adversary policy that we want the imitation learning model to train with, and let $(\widetilde{\pi}^{\nu*}, D^{\nu*})$ be an imitation learning policy and the discriminator network that are trained with adversary policy π^α. To bound the gap between $|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})|$, we consider the following two cases

- If $\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) \geq \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})$, then

$$\begin{aligned}&\left|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\right| = \phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*}) \\ &\overset{(e)}{\leq} K\epsilon + \phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^\alpha) - \min_D \phi^E(\widetilde{\pi}^{\nu*}, D|\pi^{\alpha*}) \\ &\overset{(f)}{=} K\epsilon + \min_D \phi^E(\widetilde{\pi}^{\nu*}, D|\pi^\alpha) - \min_D \phi^E(\widetilde{\pi}^{\nu*}, D|\pi^{\alpha*}) \\ &\leq K\epsilon + \left|\min_D\{\phi^E(\widetilde{\pi}^{\nu*}, D|\pi^\alpha)\} - \min_D\{\phi^E(\widetilde{\pi}^{\nu*}, D|\pi^{\alpha*})\}\right| \\ &\overset{(g)}{\leq} 2K\epsilon, \end{aligned} \tag{20}$$

  where (e) is because $|\phi^E(\widetilde{\pi}^{\nu*}, D|\pi^{\alpha*}) - \phi^E(\widetilde{\pi}^{\nu*}, D|\pi^\alpha)| \leq K\epsilon$ (according to (16)), (f) is due to $D^{\nu*} = \text{argmax}_D \phi^E(\widetilde{\pi}^{\nu*}, D|\pi^\alpha)$ and (g) is because of (19).
- If $\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) \geq \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})$, we let $\widetilde{\pi}^{\nu**} = \text{argmax}_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})$ and write

$$\begin{aligned}&\left|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})\right| = \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*}) - \phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) \\ &\overset{(h)}{\leq} \min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^{\alpha*}) - \phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^\alpha) + K\epsilon \\ &\overset{(i)}{\leq} K\epsilon + \min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^{\alpha*}) - \min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^\alpha) \\ &\leq K\epsilon + \left|\min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^{\alpha*}) - \min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^\alpha)\right| \\ &\overset{(j)}{\leq} 2K\epsilon, \end{aligned} \tag{21}$$

  where (h) is due to $\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^\alpha) - \phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) \leq K\epsilon$ (see (16)), (i) is due to $\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^\alpha) = \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^\alpha) \geq \min_D \phi^E(\widetilde{\pi}^{\nu**}, D|\pi^\alpha)$, and (j) is because of (19).

Combine the two cases above we obtain he desired bound.

$$|\phi^E(\widetilde{\pi}^{\nu*}, D^{\nu*}|\pi^{\alpha*}) - \max_{\widetilde{\pi}^\nu}\min_D \phi^E(\widetilde{\pi}^\nu, D|\pi^{\alpha*})| \leq 2K \max_{s\in\mathcal{S}}\left\{\sqrt{\text{KL}(\pi^\alpha(\cdot|s)||\widetilde{\pi}^\alpha(\cdot|s))}\right\}$$

. □

A.6 Proof of Proposition 5.1

Proposition A.6. *The gradient of* (6) *w.r.t adversary's policy can be computed as follows:*

$$\nabla_\theta \left(V_{\pi^\alpha_\theta}(s_0) - V_{\pi^\nu}(s_0|\boldsymbol{q}^\nu(\pi^\alpha_\theta)) \right) = \mathbb{E}_{\tau\sim(\pi^\nu,\pi^\alpha)} \left[\Delta^R(\tau) \sum_t \nabla_\theta \log \pi^\alpha_\theta(a^\alpha_t|s_t) \right]$$

where $\Delta^R(\tau) = \sum_t \gamma^t (r^\alpha(s_t) - r^\nu(s_t))$.

Proof. Similarly to the proof of Lemma 4.1, we compute the gradient of $V_{\pi^\nu}(\cdot)$ as

$$\begin{aligned}\nabla_\theta \left(V_{\pi^\nu}(s_0|\mathbf{q}^\nu(\pi^\alpha_\theta)) \right) &= \sum_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}\sim\pi^\alpha,\pi^\nu} \left(\sum_t \gamma^t r^\nu(s_t) \right) P(\tau) \sum_t \nabla_\theta \log \pi^\alpha_\theta(a^\alpha_t|s_t) \\ &= \mathbb{E}_{\tau=\{(s_t,a^\alpha_t,a^\nu_t)\}\sim\pi^\alpha,\pi^\nu} \left[\left(\sum_t \gamma^t r^\nu(s_t) \right) \sum_t \nabla_\psi \log \pi^\alpha_\theta(a^\alpha_t|s_t) \right],\end{aligned}$$

Thus, we can write the gradient of (6) as

$$\nabla_\theta \left(V_{\pi^\alpha_\theta}(s_0) - V_{\pi^\nu}(s_0|\mathbf{q}^\nu(\pi^\alpha_\theta)) \right) = \mathbb{E}_{\tau\sim(\pi^\nu,\pi^\alpha)} \left[\left(\sum_t \gamma^t (r^\alpha(s_t) - r^\nu(s_t)) \right) \sum_t \nabla_\theta \log \pi^\alpha_\theta(a^\alpha_t|s_t) \right],$$

which concludes the proof. □

A.7 Proof of Corollary 5.2

Corollary A.7. (6) *is equivalent to*

$$\max_{\pi^\alpha} \left\{ \mathbb{E}_{\tau\sim\pi^\alpha} \left[\sum_{t=0}^{\infty} \gamma^t \Delta^r(s_t) \; \middle| \boldsymbol{q}^\alpha(\pi^\nu) \right] \right\},$$

where $\Delta^r(s_t) = r^\alpha(s_t) - r^\nu(s_t)$.

Proof. The equivalence can be straightforwardly deduced from Proposition 5.1. □

A.8 Proof of Theorem 5.3

Theorem A.8. *For any* $\epsilon > 0$, *we have the following bound*

$$\left| Y(\epsilon) - Y^* \right| \leq \frac{\gamma\sqrt{2\ln 2}\max_s\{|\Delta^r(s)|\}}{(1-\gamma)^2}\sqrt{\epsilon}.$$

Proof. In analogy to the proofs of Lemma 4.4 and Theorem 4.5, if we define $\Lambda(\pi^\nu)$ as the adversary's expected return if the victim's policy is π^ν

$$\Lambda(\pi^\nu) = \max_{\pi^\alpha} \left\{ \mathbb{E}_{\tau\sim\pi^\alpha} \left[\sum_{t=0}^{\infty} \gamma^t \Delta^r(s_t) \; \middle| \mathbf{q}^\alpha(\pi^\nu) \right] \right\}$$

then, similarly to the derivation in (15), we can get the following bound for any $\pi^\nu \in \Omega(\epsilon)$

$$\begin{aligned}\left|\Lambda(\pi^\nu) - \Lambda(\pi^\nu_0)\right| &\leq \frac{\gamma \max_s |\Delta^r(s)|}{(1-\gamma)^2} \left\{ \sqrt{2\ln 2\mathrm{KL}(\pi^\nu(\cdot|s)||\widetilde{\pi}^\nu(\cdot|s))} \right\} \\ &\leq \frac{\sqrt{2\ln 2\epsilon}\gamma \max_s |\Delta^r(s)|}{(1-\gamma)^2} \qquad (22)\end{aligned}$$

which also implies that

$$\left| \min_{\pi^\nu\in\Omega(\epsilon)} \{\Lambda(\pi^\nu)\} - Y^* \right| \leq \frac{\sqrt{2\ln 2\epsilon}\gamma \max_s |\Delta^r(s)|}{(1-\gamma)^2},$$

which is also the desired result. □

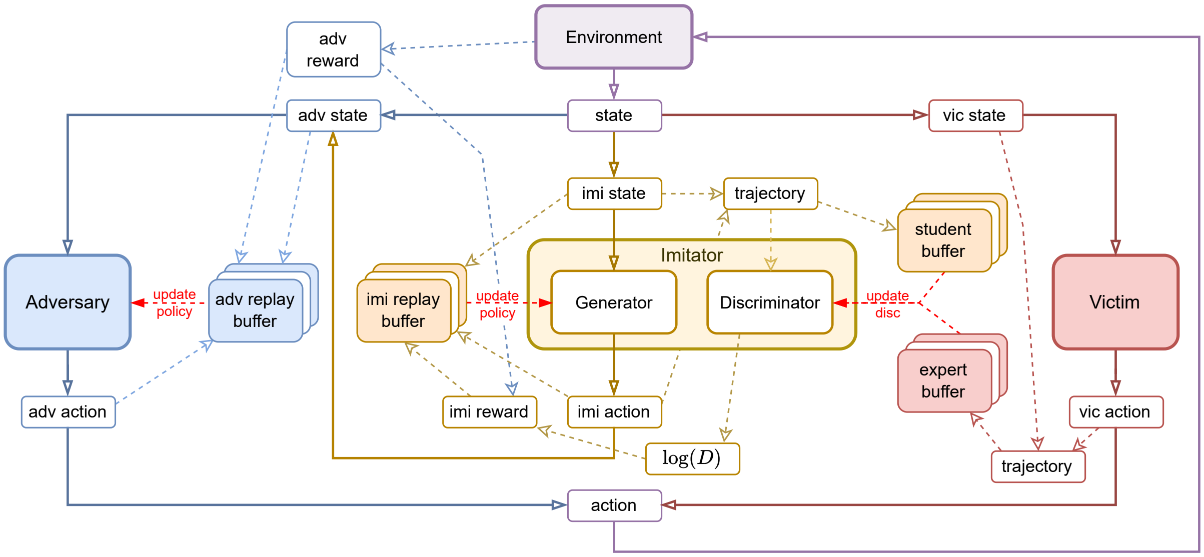
B ADVERSARIAL POLICY IMITATION LEARNING ALGORITHM

Algorithm 2 shows the detailed steps of our Adversarial Policy Imitation Learning algorithms and Figure 5 provides a more detailed overview of our framework.

Algorithm 2 Adversarial Policy Imitation Learning

```
Input: Environment $env$; adversary policy $\pi_\theta^\alpha$; defender policy $\pi^\nu$; imitator defender policy $\widetilde{\pi}_\psi^\nu$; discriminator $D_w$; adversary replay buffer $\mathcal{D}^\alpha$; imitator replay buffer $\widetilde{\mathcal{D}}^\nu$; expert trajectory buffer $\mathcal{D}_\tau^\nu$; imitator trajectory buffer $\widetilde{\mathcal{D}}_\tau^\nu$; initial trainable parameters $\theta_0, \psi_0, w_0$.
while repeat a certain number of times do
    $s_0 \leftarrow env.\text{reset}()$                                    ▷ Start new episode
    for $t = 0, 1, 2, \ldots$ do
        # Sampling
        Sample victim action $a_t^\nu$ and imitator action $\widetilde{a}_t^\nu$ by $\pi^\nu(s_t)$ and $\widetilde{\pi}_\psi^\nu(s_t)$.
        Sample adversary action $a_t^\alpha \sim \pi_\theta^\alpha(s_t')$, where $s_t' = (s_t, \widetilde{a}_t^\nu)$.     ▷ Sample actions
        $(s_{t+1}, r_t, d_t) \leftarrow env.\text{step}(a_t^\nu, a_t^\alpha)$                   ▷ Next step
        $\widetilde{r}_t^\nu \leftarrow \eta(s_t, a_t^\alpha, a_t^\nu)$
        Append transition $(s_t', a_t^\alpha, r_t^\alpha)$, $(s_t, \widetilde{a}_t^\nu, \widetilde{r}_t^\nu)$, $(s_t, a_t^\nu)$ and $(s_t, \widetilde{a}_t^\nu)$ respectively to $\mathcal{D}^\alpha$, $\widetilde{\mathcal{D}}^\nu$, $\mathcal{D}_\tau^\nu$ and $\widetilde{\mathcal{D}}_\tau^\nu$.
        if any buffer is full then                                      ▷ Update parameters
            for $i = 0, 1, 2, \ldots$ do
                Sample trajectories from $\mathcal{D}^\alpha$, $\widetilde{\mathcal{D}}^\nu$, $\mathcal{D}_\tau^\nu$ and $\widetilde{\mathcal{D}}_\tau^\nu$
                # Updating imitator's discriminator parameters
                Update imitator's discriminator parameters from $w_i$ to $w_{i+1}$ using (4).
                Take an imitator's generator policy step from $\psi_i$ to $\psi_{i+1}$ using TRPO rule using gradients in (5)
                # Updating adversary's policy network
                Take an adversary policy step from $\theta_i$ to $\theta_{i+1}$ via PPO using the gradients given in (5.1).
            end for
            Clear all buffers and update trainable parameters $\theta_0, \psi_0, w_0$.
        end if
        if $d_t$ is true then                                           ▷ Terminated
            break
        end if
    end for
end while
```

Figure 5: An overview of our imitation-based adversarial policy learning (APIL) platform.



C EXPERIMENTAL SETTINGS AND ADDITIONAL EXPERIMENTS

Similar to APL's settings [8], we train our adversary agents with 35M steps and retrain victim with 10M steps. In addition, we use the same model architectures, hidden dimensions and hyperparameters with 1 GPU Nvidia GeForce RTX 2080 Ti, 24-core CPU Intel Xeon 4116 @ 2.1GHz and 64G RAM. All evaluation are tested individually with different seeds over 1000 episodes.

When retraining the victim, we let the victim agent play with a mixing adversary which is a combination of the newly trained adversary and the baseline one. That is, we randomly taking actions from both adversaries. In Figures 6 and 7, we report the performance of the retraining victim, but with each adversary separately. It is clear that the victim retraining performance improves over episodes for both adversaries for *Kick-and-Defend, You-Shall-Not-Pass* and *Sumo-Humans*. However, for *Sumo-Ants*, the performance from retraining with our trained adversary improves, but that from retraining with the baseline adversary degrades. This also indicates an advantage of our trained adversary agent in terms of retraining the victim.

Figure 6: Performance of retraining victim with trained adversary.
First line: win-rate. Second line: win-rate + tie-rate.

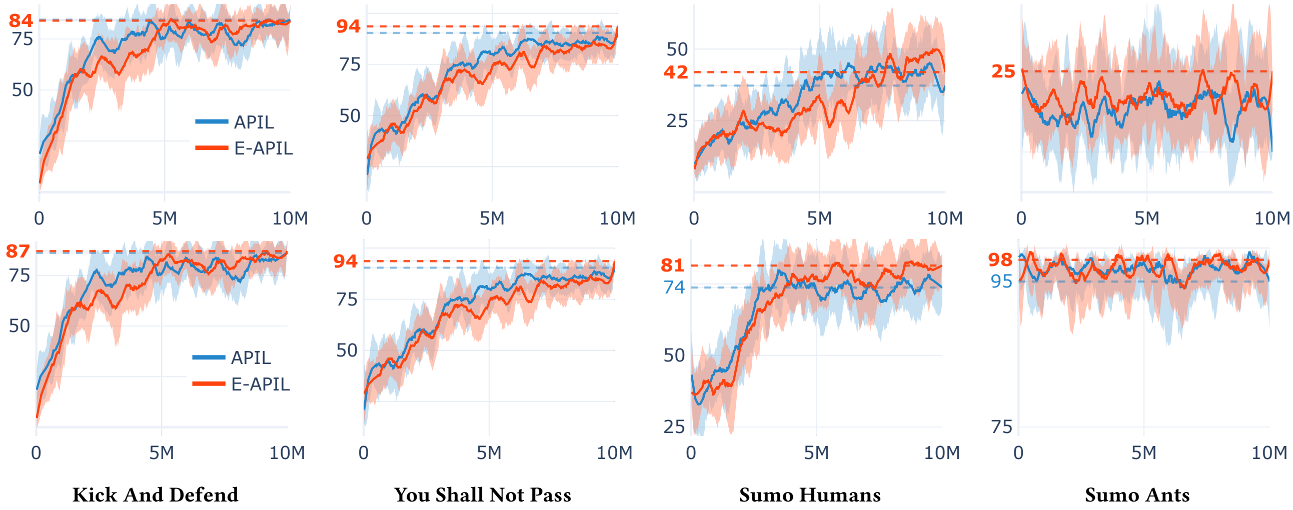
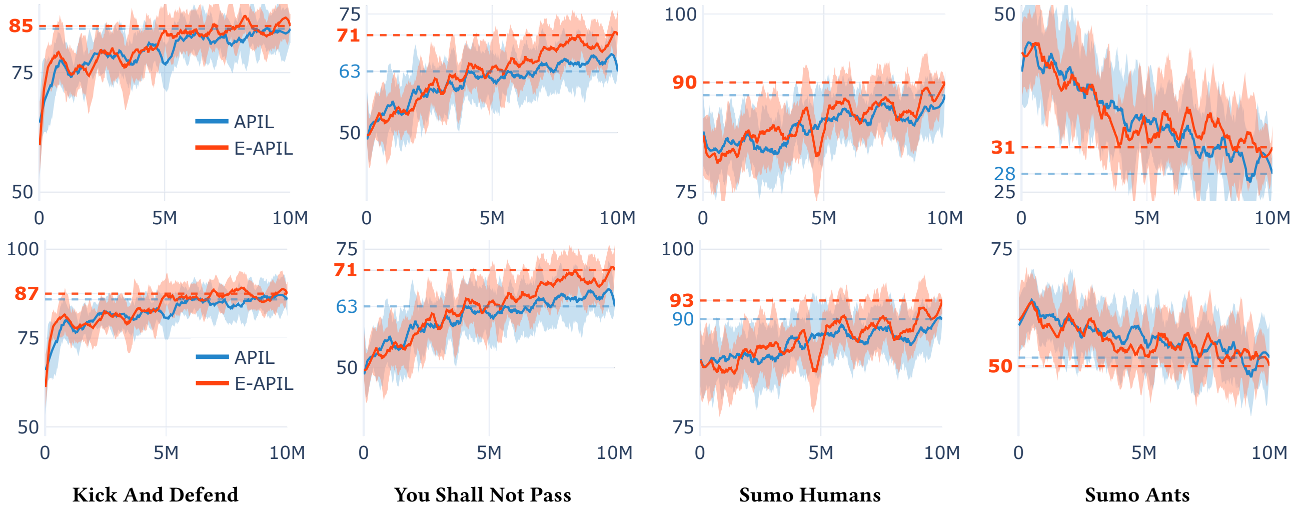


Figure 7: Performance of retraining victim with baseline adversary.
First row: win-rate. Second row: win-rate + tie-rate.


APIL
E-APIL
5M
10M
APIL
E-APIL
Kick And Defend
You Shall Not Pass
Sumo Humans
Sumo Ants