RAT: Adversarial Attacks on Deep Reinforcement Agents for Targeted Behaviors

Fengshuo Bai[1, 2], **Runze Liu**[3],
Yali Du[4], **Ying Wen**[1, *], **Yaodong Yang**[5, 6, *]
[1]Shanghai Jiao Tong University
[2]Zhongguancun Academy
[3]Tsinghua Shenzhen International Graduate School, Tsinghua University
[4]King's College London
[5]Center for AI Safety and Governance, Institute for AI, Peking University
[6]State Key Laboratory of General Artificial Intelligence, Peking University
*{yaodong.yang@pku.edu.cn, ying.wen@sjtu.edu.cn}

Abstract

Evaluating deep reinforcement learning agents against targeted behavior attacks is critical for assessing their robustness. These attacks aim to manipulate the victim into specific behaviors that align with the attacker's objectives, often bypassing traditional reward-based defenses. Prior methods have primarily focused on reducing cumulative rewards; however, rewards are typically too generic to capture complex safety requirements effectively. As a result, focusing solely on reward reduction can lead to suboptimal attack strategies, particularly in safety-critical scenarios where more precise behavior manipulation is needed. To address these challenges, we propose RAT, a method designed for universal, targeted behavior attacks. RAT trains an intention policy that is explicitly aligned with human preferences, serving as a precise behavioral target for the adversary. Concurrently, an adversary manipulates the victim's policy to follow this target behavior. To enhance the effectiveness of these attacks, RAT dynamically adjusts the state occupancy measure within the replay buffer, allowing for more controlled and effective behavior manipulation. Our empirical results on robotic simulation tasks demonstrate that RAT outperforms existing adversarial attack algorithms in inducing specific behaviors. Additionally, RAT shows promise in improving agent robustness, leading to more resilient policies. We further validate RAT by guiding Decision Transformer agents to adopt behaviors aligned with human preferences in various MuJoCo tasks, demonstrating its effectiveness across diverse tasks.

1 Introduction

Reinforcement learning (RL) (Sutton and Barto 2018) combined with deep neural networks (DNN) (LeCun, Bengio, and Hinton 2015) shows extraordinary capabilities of allowing agents to master complex behaviors in various domains, including robotic manipulation (Wang et al. 2023; Bai et al. 2023), video games (Zhang et al. 2023, 2024b; Wang* et al. 2024; Wen et al. 2024; Zhang et al. 2025), industrial applications (Xu and Yu 2023; Shi et al. 2024; Jia et al. 2024). However, recent findings (Huang et al. 2017; Pattanaik et al. 2018; Zhang et al. 2020, 2024a) show that even well-trained DRL agents suffer from vulnerability against test-time attacks, raising concerns in high-risk or safety-critical situations. To understand adversarial attacks on learning algorithms and enhance the robustness of DRL agents, it is crucial to evaluate the performance of the agents under any potential adversarial attacks with certain constraints. In other words, identifying a universal and strong adversary is essential.

Figure 1: An example illustrating the distinction between our approach and generic attacks.

Existing methods pay little attention to devising universal, efficient, targeted behavior attacks. Firstly, several methods primarily focused on reducing the cumulative reward often lack specified attack targets. Prior research (Zhang et al. 2020, 2021; Sun et al. 2022) considers training strong adversaries by perturbing state observations of victims to achieve the worst-case expected return. However, rewards lack the expressiveness to adequately encode complex safety requirements (Vamplew et al. 2022; Hasanbeig, Kroening, and Abate 2020). Additionally, requiring the victim's training rewards to craft such attacks is generally impractical. Therefore, only quantifying the decrease in cumulative reward can be too generic and result in suboptimal attack performance, particularly when adversaries are intended to execute specific safety-related attacks. Consider the scenario depicted in Figure 1, where a robot's objective is to collect coins. Previous attack methods aim at inducing the robot away from the coins by minimizing its expected return. However, this approach overlooks specific unsafe behaviors, such as manipulating the robot to collide with a bomb. Secondly, the previous targeted attack only considered predefined targets, which resulted in rigidity and inefficiency. (Hussenot, Geist, and Pietquin 2019a; Lin et al. 2017a) mainly focuses on misleading the agent towards a predetermined state or target policy,

overlooking specific behaviors. Additionally, the difficulty of providing a well-designed targeted policy makes these methods hard to apply. In a broader context, these adversarial attacks are incapable of controlling the behaviors of agents as a form of universal attack.

In this paper, we present a novel adversarial attack method, RAT, which focuses on Adve**R**sarial **A**ttacks against deep reinforcement learning agents for **T**argeted behavior. RAT consists of three core components: an intention policy, an adversary, and a weighting function, all trained simultaneously. Unlike previous methods that rely on predefined target policies, RAT dynamically trains an intention policy that aligns with human preferences, providing a flexible and adaptive behavioral target for the adversary. By leveraging advances in preference-based reinforcement learning (PbRL) (Lee, Smith, and Abbeel 2021; Park et al. 2022; Liu et al. 2022; Bai et al. 2024), the intention policy efficiently captures human intent during the training process. RAT employs the adversary to perturb the victim agent's observations, guiding the agent towards the behaviors specified by the intention policy. To further enhance attack effectiveness, we introduce a weighting function that adjusts the state occupancy measure, optimizing the distribution of states visited during training. This adjustment improves both the performance and efficiency of the attack. Through iterative refinement, RAT steers the victim agent toward specific human-desired behaviors with greater precision than existing adversarial attack methods.

Our contributions are summarized as follows: **(1)** We propose a universal targeted behavior attack method against DRL agents, designed to induce specific behaviors in a victim agent across a wide range of tasks. **(2)** We provide a theoretical analysis of RAT, offering a convergence guarantee under clearly defined conditions, which enhances the understanding of its effectiveness. **(3)** Through extensive experiments across various domains, we demonstrate that RAT significantly outperforms existing adversarial attack methods, showing that both online and offline RL agents, including Decision Transformer, are susceptible to our approach. **(4)** We introduce two variants, RAT-ATLA and RAT-WocaR, which demonstrate how RAT can be effectively employed to enhance the robustness of DRL agents through adversarial training, showing its versatility in both attack and defense.

2 Related Work

Adversarial Attacks on State Observations in DRL. Huang et al. (2017) applies the Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) to compute adversarial perturbations, directing the victim policy towards suboptimal actions. Pattanaik et al. (2018) introduces a strategy to make the victim choose the worst action based on its Q-function. Gleave et al. (2020) focuses on adversarial attacks within the context of a two-player Markov game rather than altering the agent's observation. Zhang et al. (2020) proposes the state-adversarial MDP (SA-MDP) and develops two adversarial attack methods, Robust Sarsa (RS) and Maximal Action Difference (MAD). SA-RL (Zhang et al. 2021) optimizes an adversary to perturb states using end-to-end RL. PA-AD (Sun et al. 2022) utilizes an RL-based "director" to determine the best policy perturbation direction and an optimization-based "actor" to generate perturbed states accordingly. Another line of work focuses on steering DRL agents toward specific states or policies. Lin et al. (2017b); Buddareddygari et al. (2022) propose targeted adversarial attack methods against DRL agents, aimed at directing the agent to a specific state. Hussenot, Geist, and Pietquin (2019b) offer a novel approach by attacking the agent to mimic a target policy. However, these methods often require access to the victim's training reward or a predetermined target state or policy, which may be impractical. Our method differs from these methods by emphasizing the manipulation of the victim's behaviors without needing access to the victim's training reward or a pre-defined target state or policy.

Robustness for State Observations in DRL. Training DRL agents with perturbed state observations from adversaries has been explored in various studies. Shen et al. (2020); Oikarinen et al. (2021) focus on a strategy, ensuring that the policy produces similar outputs for similar inputs, which has demonstrated certifiable performance in video games. Another research direction, as presented in Pinto et al. (2017); Mandlekar et al. (2017); Pattanaik et al. (2018), aims to enhance an agent's robustness by training it under adversarial attacks. Zhang et al. (2021) proposes ATLA, a method that alternates between training an RL agent and an RL adversary, significantly enhancing policy robustness. Building on this concept, Sun et al. (2022) proposed PA-ATLT, which employs a similar approach but utilizes a more advanced RL attacker. And several methods proposed by Fischer et al. (2019); Lütjens, Everett, and How (2020), concentrate on the lower bounds of the Q-function to certify an agent's robustness at every step. WocaR-RL (Liang et al. 2022b) is an efficient method that directly estimates and optimizes the worst-case reward of a policy under attacks without requiring extra samples for learning an attacker.

Preference-based RL. PbRL provides an effective way to incorporate human preferences into agent learning. Christiano et al. (2017) proposes a foundational framework for PbRL. Ibarz et al. (2018) utilizes expert demonstrations to initialize the policy, besides learning the reward model from human preferences. Nonetheless, these earlier methods often require extensive human feedback, which is typically not feasible in practical scenarios. Recent studies have addressed this limitation: Lee, Smith, and Abbeel (2021) develops a feedback-efficient PbRL algorithm, leveraging unsupervised exploration and reward relabeling. Park et al. (2022) furthers feedback efficiency through semi-supervised reward learning and data augmentation. Meanwhile, Liang et al. (2022a) proposes an intrinsic reward to enhance exploration. Continuing this trend, Liu et al. (2022) improves feedback efficiency by aligning the Q-function with human preferences. Additionally, several works (Bai et al. 2024; Liu et al. 2024) have been dedicated to improving feedback efficiency by providing diverse insights. In our research, we employ PbRL to capture human intent and train an intention policy, which serves as the learning target for training adversaries.

3 Problem Setup and Notations

The Victim Policy. In RL, agent learning can be modeled as a finite-horizon Markov Decision Process (MDP) defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$. $\mathcal{S}$ and $\mathcal{A}$ denote state and action space, respectively. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the transition dynamics, which determines the probability of transferring to $\mathbf{s}'$ given state $\mathbf{s}$ and action $\mathbf{a}$. We denote the stationary policy $\pi_\nu : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, where ν are parameters of the victim. We suppose the victim policy is fixed and uses the approximator.

Threat Model. To study targeted behavior attack with human preferences, we formulate it as rewarded state-adversarial Markov Decision Process (RSA-MDP). Formally, a RSA-MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{B}, \widehat{\mathcal{R}}, \mathcal{P}, \gamma)$. The adversary $\pi_\alpha : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ perturbs the states before the victim observes them, where α are parameters of the adversary. The adversary perturbs the state $\mathbf{s}$ into $\tilde{\mathbf{s}}$ restricted by $\mathcal{B}(\mathbf{s})$ (i.e., $\tilde{\mathbf{s}} \in \mathcal{B}(\mathbf{s})$). $\mathcal{B}(\mathbf{s})$ is defined as a small set $\{\tilde{\mathbf{s}} \in \mathcal{S} : \| \mathbf{s} - \tilde{\mathbf{s}} \|_p \leq \epsilon\}$, which limits the attack power of the adversary, and ϵ is the attack budget. Since directly generating $\tilde{\mathbf{s}} \in \mathcal{B}(\mathbf{s})$ is hard, the adversary learns to produce a Gaussian noise Δ with $\ell_\infty(\Delta)$ less than 1, and we obtain the perturbed state through $\tilde{\mathbf{s}} = \mathbf{s} + \Delta * \epsilon$. The victim takes action according to the observed $\tilde{\mathbf{s}}$, while true states in the environment are not changed. Recall that $\pi_{\nu \circ \alpha}$ denotes the perturbed policy caused by adversary π_α, i.e., $\pi_{\nu \circ \alpha}(\cdot|\mathbf{s}) = \pi_\nu(\cdot|\pi_\alpha(\mathbf{s})), \forall \mathbf{s} \in \mathcal{S}$.

Unlike SA-MDP (Zhang et al. 2020), RSA-MDP introduces $\widehat{\mathcal{R}}$, which learns from human preferences. The target of RSA-MDP is to solve the optimal adversary π^*_α, which enables the victim to achieve the maximum cumulative reward (i.e., from $\widehat{\mathcal{R}}$) over all states. Lemma 3.1 shows that solving the optimal adversary in RSA-MDP is equivalent to finding the optimal policy in MDP $\hat{\mathcal{M}} = (\mathcal{S}, \hat{\mathcal{A}}, \widehat{\mathcal{R}}, \widehat{\mathcal{P}}, \gamma)$, where $\hat{\mathcal{A}} = \mathcal{S}$ and $\widehat{\mathcal{P}}$ is the transition dynamics of the adversary.

Lemma 3.1. *Given a RSA-MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{B}, \widehat{\mathcal{R}}, \mathcal{P}, \gamma)$ and a fixed victim policy π_ν, there exists a MDP $\hat{\mathcal{M}} = (\mathcal{S}, \hat{\mathcal{A}}, \widehat{\mathcal{R}}, \widehat{\mathcal{P}}, \gamma)$ such that the optimal policy of $\hat{\mathcal{M}}$ is equivalent to the optimal adversary π_α in RSA-MDP given a fixed victim. Here, $\hat{\mathcal{A}} = \mathcal{S}$, and for $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$ and $\hat{\mathbf{a}} \in \hat{\mathcal{A}}$,*

$$\widehat{\mathcal{P}}(\mathbf{s}'|\mathbf{s}, \hat{\mathbf{a}}) = \sum_{\mathbf{a} \in \mathcal{A}} \pi_\nu(\mathbf{a}|\hat{\mathbf{a}}) \mathcal{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a}).$$

4 Method

In this section, we introduce RAT, a generic framework adaptable to any RL algorithm for conducting targeted behavior attack against DRL learners. RAT is composed of three integral components: an intention policy π_θ, the adversary π_α, and the weighting function h_ω, all of which are trained in tandem. The fundamental concept behind RAT is twofold: **(1)** It develops an intention policy to serve as the learning objective for the adversary. **(2)** A weighting function is trained to adjust the state occupancy measure of replay buffer, and the training of π_α and h_ω is formulated as a bi-level optimization problem. The framework of RAT is depicted in Figure 3.

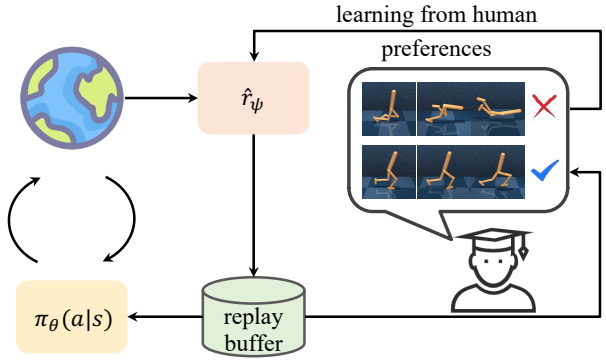


Figure 2: **Diagram of PbRL.** The reward model $\hat{r}_\psi$ is trained to align with human intention, providing estimations of rewards for policy learning. The policy is optimized by using transitions relabeled by the up-to-date reward model.

4.1 Learning Intention Policy

RAT is designed to find an optimal adversary capable of manipulating the victim's behaviors in alignment with human intentions. To achieve this, we consider capturing human intentions and training an intention policy π_θ, which translates these abstract intentions into action-level behaviors. A practical approach to realizing this concept is through PbRL, a method that aligns the intention policy with human intent without the need for reward engineering. As depicted in Figure 2, within the PbRL framework, the agent does not rely on a ground-truth reward function. Instead, humans provide preference labels comparing two agent trajectories, and the reward model $\hat{r}_\psi$ is trained to match human preferences (Christiano et al. 2017).

Formally, we denote a state-action sequence of length k, $\{\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \cdots, \mathbf{s}_{t+k}, \mathbf{a}_{t+k}\}$ as a segment σ. Given a pair of segments (σ^0, σ^1), humans provide a preference label y indicating which segment is preferred. Here, y represents a distribution, specifically $y \in \{(0, 1), (1, 0), (0.5, 0.5)\}$. In accordance with the Bradley-Terry model (Bradley and Terry 1952), we construct a preference predictor as shown in (1):

$$P_\psi[\sigma^0 \succ \sigma^1] = \frac{\exp \sum_t \hat{r}_\psi(\mathbf{s}^0_t, \mathbf{a}^0_t)}{\sum_{i \in \{0,1\}} \exp \sum_t \hat{r}_\psi(\mathbf{s}^i_t, \mathbf{a}^i_t)}, \quad (1)$$

where $\sigma^0 \succ \sigma^1$ indicates a preference for σ^0 over σ^1. This predictor determines the probability of a segment being preferred, proportional to its exponential return.

The reward model is optimized to align the predicted preference labels with human preferences using a cross-entropy loss, as expressed in the following equation:

$$\mathcal{L}(\psi) = - \mathop{\mathbb{E}}_{(\sigma^0, \sigma^1, y) \sim \mathcal{D}} \Big[\sum_{i=0}^{1} y(i) \log P_\psi[\sigma^i \succ \sigma^{1-i}] \Big], \quad (2)$$

where $\mathcal{D}$ represents a dataset of triplets (σ^0, σ^1, y) that consist of segment pairs and corresponding human preference labels. By minimizing the cross-entropy loss as defined in (2), we derive an estimated reward function $\hat{r}_\psi$. This function is then utilized to provide reward estimations for policy

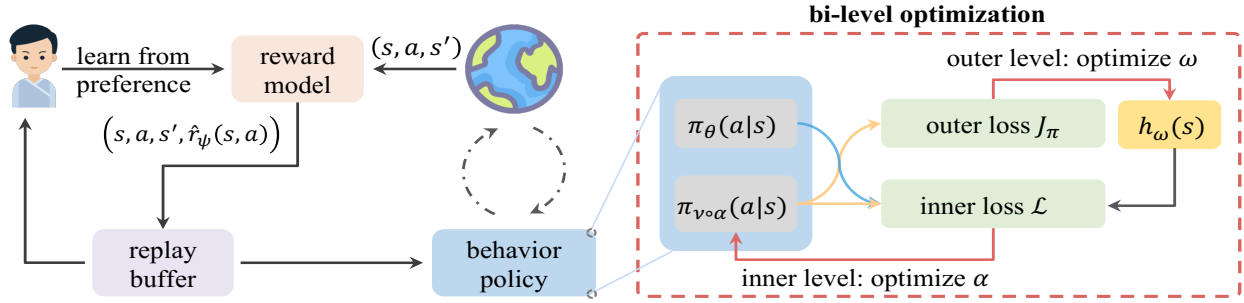


Figure 3: **Overview of RAT.** During training, it learns the intention policy π_θ and the reward model $\widehat{r}_\psi$, following the principles of PbRL. Simultaneously, it trains an adversary π_α and a weighting function h_ω within a bi-level optimization framework. **In the inner-level**, the adversary is optimized such that the perturbed policy aligns with the intention policy. A validation loss J_π is introduced, serving as a metric to assess the adversary's performance. **In the outer-level**, the weighting function is updated to improve the performance of the adversary by minimizing the outer loss J_π.

learning using any RL algorithm. Following PEBBLE (Lee, Smith, and Abbeel 2021), we employ the Soft Actor-Critic (SAC) (Haarnoja et al. 2018) algorithm to train the intention policy π_θ. The Q-function Q_ϕ is optimized by reducing the Bellman residual, as defined below:

$$J_Q(\phi) = \mathop{\mathbb{E}}_{\tau_t \sim \mathcal{B}} \left[\left(Q_\phi(\mathbf{s}_t, \mathbf{a}_t) - \widehat{r}_t - \gamma \bar{V}(\mathbf{s}_{t+1}) \right)^2 \right], \quad (3)$$

where $\bar{V}(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta} \left[Q_{\bar{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \mu \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \right]$, $\tau_t = (\mathbf{s}_t, \mathbf{a}_t, \widehat{r}_t, \mathbf{s}_{t+1})$ represents the transition at time t, with $\bar{\phi}$ being the parameter of the target soft Q-function. The intention policy π_θ is updated to minimize the following loss:

$$J_\pi(\theta) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{B}, \mathbf{a}_t \sim \pi_\theta} \left[\mu \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) - Q_\phi(\mathbf{s}_t, \mathbf{a}_t) \right], \quad (4)$$

where μ is the temperature parameter.

In this way, RAT effectively captures human intent via the reward model $\widehat{r}_\psi$ and leverages π_θ to provide behavior-level guidance for the training of the adversary.

4.2 Learning Adversary and Weighting Function

To steer the victim policy towards behaviors desired by humans, RAT trains the adversary by minimizing the Kullback-Leibler (KL) divergence between the perturbed policy $\pi_{\nu \circ \alpha}$ and the intention policy π_θ. Additionally, certain pivotal moments during adversary training can significantly influence the success rate of attacks. To ensure a stable training process and enhance the adversary's performance, a weighting function h_ω is introduced to re-weight the state occupancy measure of dataset.

Formally, our method is formulated as a bi-level optimization algorithm. It alternates between updating the adversary π_α and the weighting function h_ω through inner and outer optimization processes. In the inner level, the adversary's parameters α are optimized by minimizing the re-weighted KL divergence between $\pi_{\nu \circ \alpha}$ and π_θ, as specified in (6). At the outer level, the weighting function is developed to identify crucial states and improve the adversary's performance, as guided by a performance metric of the adversary. This metric is represented as a meta-level loss J_π, detailed in (7). The whole objective of RAT is formulated as:

$$\begin{aligned} \min_{\omega} \quad & J_\pi(\alpha(\omega)), \\ \text{s.t.} \quad & \alpha(\omega) = \arg\min_{\alpha} \mathcal{L}(\alpha; \omega, \theta). \end{aligned} \quad (5)$$

Inner Loop: Training Adversary π_α. In inner-level optimization, with the given intention policy π_θ and the weighting function h_ω, the goal is to identify the optimal adversary. This is achieved by minimizing the re-weighted KL divergence between $\pi_{\nu \circ \alpha}$ and π_θ, as shown in equation (6):

$$\mathcal{L}(\alpha; \omega, \theta) = \mathop{\mathbb{E}}_{\mathbf{s} \sim \mathcal{B}} \left[h_\omega(\mathbf{s}) D_{\mathrm{KL}} \left(\pi_{\nu \circ \alpha}(\cdot|\mathbf{s}) \parallel \pi_\theta(\cdot|\mathbf{s}) \right) \right], \quad (6)$$

where $h_\omega(\mathbf{s})$ represents the importance weights determined by the weighting function h_ω.

Intuitively, the adversary is optimized to ensure that the perturbed policy $\pi_{\nu \circ \alpha}$ aligns behaviorally with the intention policy. Concurrently, h_ω allocates varying weights to states, reflecting their differing levels of importance. Through the synergistic effort of the intention policy and the weighting function, our method effectively trains an optimal adversary.

Outer Loop: Training Weighting Function h_ω. In outer-level optimization, the goal is to develop a precise weighting function that can identify significant moments and refine the state occupancy measure of the replay buffer to enhance adversary learning. As the intention policy is the target for the perturbed policy, it becomes simpler to establish a validation loss. This loss measures the perturbed policy's performance and simultaneously reflects the adversary's effectiveness. Consequently, the weighting function is trained to differentiate the importance of states by optimizing this validation loss. The perturbed policy $\pi_{\nu \circ \alpha}$ is assessed using a policy loss in (7), adapted from the policy loss in (4):

$$J_\pi(\omega) = \mathop{\mathbb{E}}_{\substack{\mathbf{s}_t \sim \mathcal{B} \\ \mathbf{a}_t \sim \pi_{\nu \circ \alpha(\omega)}}} \left[\mu \log \pi_{\nu \circ \alpha(\omega)}(\mathbf{a}_t|\mathbf{s}_t) - Q_\phi(\mathbf{s}_t, \mathbf{a}_t) \right], \quad (7)$$

where $\alpha(\omega)$ denotes α implicitly depends on ω. The optimization process involves calculating the implicit derivative of $J_\pi(\alpha(\omega))$ with respect to ω and finding the optimal ω^* through optimization.

	Dataset	PA-AD (oracle)	PA-AD	SA-RL (oracle)	SA-RL	Random	RAT (ours)
Mani	Door Lock	4.50 ± 4.00	3.50 ± 6.63	76.50 ± 14.97	39.50 ± 30.48	0.00 ± 0.00	87.00 ± 10.00
	Window Close	0.00 ± 0.00	0.50 ± 0.00	99.00 ± 3.00	23.50 ± 37.22	0.00 ± 0.00	72.50 ± 40.01
	Drawer Close	0.00 ± 0.00	0.00 ± 0.00	57.50 ± 18.00	4.00 ± 8.00	0.00 ± 0.00	76.00 ± 24.98
	Faucet Close	0.00 ± 0.00	0.00 ± 0.00	66.50 ± 16.85	4.50 ± 9.22	0.00 ± 0.00	91.00 ± 6.71
Opposite	Door Lock	9.50 ± 7.48	10.00 ± 9.17	8.00 ± 13.42	2.00 ± 0.00	1.00 ± 3.00	99.00 ± 3.00
	Window Close	38.50 ± 23.69	55.00 ± 14.70	63.00 ± 34.70	20.00 ± 39.80	5.50 ± 5.00	99.00 ± 0.00
	Drawer Close	88.50 ± 7.81	79.00 ± 18.44	81.00 ± 20.88	63.00 ± 32.50	0.00 ± 0.00	92.00 ± 17.32
	Faucet Close	19.00 ± 13.27	32.00 ± 11.00	7.00 ± 12.81	8.00 ± 16.00	0.50 ± 0.00	96.00 ± 12.81

Table 1: The average attack success rate and standard deviation are computed for different attacks targeting victim agents on Manipulation and Opposite scenarios. The results are averaged over 30 episodes.

Practical Implementation. A one-step gradient update is used to approximate $\arg\min_\alpha$, as shown in (8), thus establishing a connection between α and ω:

$$\hat{\alpha}(\omega) \approx \alpha_t - \eta_t \, \nabla_\alpha \mathcal{L}(\alpha; \omega, \theta)|_{\alpha_t} . \tag{8}$$

The gradient of the outer loss with respect to ω is then determined using the chain rule:

$$\nabla_\omega J_\pi(\alpha(\omega))|_{\omega_t} = \sum_{\mathbf{s}} f(\mathbf{s}) \cdot \nabla_\omega h(\mathbf{s})|_{\omega_t} , \tag{9}$$

where $f(\mathbf{s}) = -\eta_t \cdot (\nabla_{\hat{\alpha}} J_\pi(\alpha(\omega)))^\top \nabla_\alpha D_{\mathrm{KL}}(\pi_{\nu \circ \alpha}(\cdot|\mathbf{s}) \parallel \pi_\theta(\cdot|\mathbf{s}))$. The essence of this step is to establish and compute the relationship between α and ω. By obtaining the implicit derivative, RAT updates the parameters of the weighting function using gradient descent with an outer learning rate.

4.3 Theoretical Analysis

We provide convergence guarantee of RAT. In Theorem 4.1, we demonstrate that the gradient of the outer loss with respect to ω will converge to zero. Consequently, RAT learns a more effective adversary by leveraging the importance of the weights generated by the optimal weighting function. Theorem 4.2 addresses the convergence of the inner loss. We prove that the inner loss of RAT converges to critical points under certain reasonable conditions, thereby ensuring that the parameters of the adversary can converge towards the optimal parameters.

Theorem 4.1. *Suppose J_π is Lipschitz-smooth with constant L, the gradient of J_π and $\mathcal{L}$ is bounded by ρ. Let the training iterations be T, the inner-level optimization learning rate $\eta_t = \min\{1, \frac{c_1}{T}\}$ for some constant $c_1 > 0$ where $\frac{c_1}{T} < 1$. Let the outer-level optimization learning rate $\beta_t = \min\{\frac{1}{L}, \frac{c_2}{\sqrt{T}}\}$ for some constant $c_2 > 0$ where $c_2 \leq \frac{\sqrt{T}}{L}$, and $\sum_{t=1}^{\infty} \beta_t \leq \infty, \sum_{t=1}^{\infty} \beta_t^2 \leq \infty$. The convergence rate of J_π achieves*

$$\min_{1 \leq t \leq T} \mathbb{E}\left[\|\nabla_\omega J_\pi(\alpha_{t+1}(\omega_t))\|^2 \right] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right). \tag{10}$$

Theorem 4.2. *Under the conditions specified in Theorem 4.1, RAT achieves:*

$$\lim_{t \to \infty} \mathbb{E}\left[\|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t)\|^2 \right] = 0. \tag{11}$$

5 Experiments

In this section, we evaluate all method on a range of robotic manipulation tasks from Meta-world (Yu et al. 2020) and locomotion tasks from MuJoCo (Todorov, Erez, and Tassa 2012). Our objective is to address the following key questions: **(1)** Does our method have the capacity to implement universal targeted behavior attack against DRL learners? **(2)** Can our approach successfully deceive a commonly used offline RL method, such as the Decision Transformer (Chen et al. 2021), to execute specific behaviors? **(3)** Does our method contribute to enhancing an agent's **robustness** through adversarial training? **(4)** Are the individual components within our approach **effective**? The responses to problems $(1) - (4)$ are addressed in Sections 5.2 through 5.5, respectively.

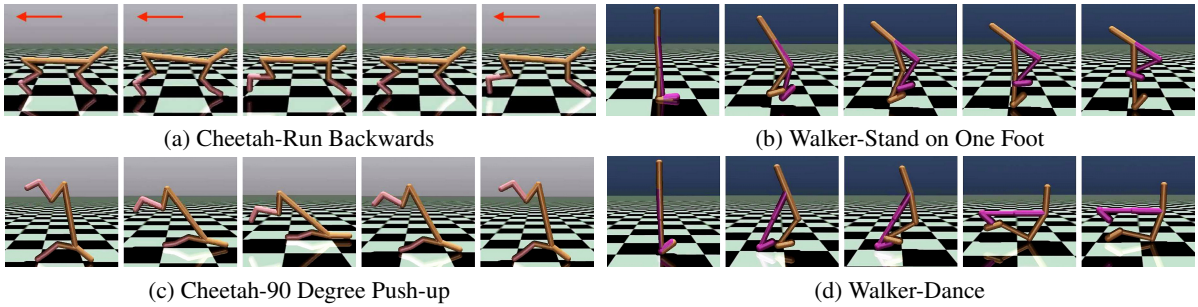
5.1 Setup

Baselines. We compare RAT with Random attack and two state-of-the-art evasion attack methods, including (1) *Random*: a basic baseline that samples random perturbed observations via a uniform distribution. (2) *SA-RL* (Zhang et al. 2021): learning an adversary in the form of end-to-end RL formulation. (3) *PA-AD* (Sun et al. 2022): combining RL-based "director" and non-RL "actor" to find state perturbations.

Implementation Settings. In our experiments, all methods follow PEBBLE (Lee, Smith, and Abbeel 2021) to learn the reward model using the same number of preference labels. The key modification in employing PbRL is that the rewards in transitions are derived from the reward model $\widehat{r}_\psi$, rather than ground-truth rewards, and this model is trained by minimizing (2). Specifically, in the original versions of SA-RL (Zhang et al. 2021) and PA-AD (Sun et al. 2022), the negative value of the reward obtained by the victim is used to train adversaries. We adapt this by using estimated rewards from $\widehat{r}_\psi$. To evaluate performance effectively and expedite the training, we follow the foundational settings in PbRL (Lee, Smith, and Abbeel 2021; Park et al. 2022; Liu et al. 2022), considering the use of a scripted teacher that always provides accurate preference labels. Moreover, to minimize the influence of PbRL, we include oracle versions of SA-RL and PA-AD, which utilize the ground-truth rewards of the targeted task. For implementing SA-RL[1] and PA-AD[2],

[1]https://github.com/huanzhang12/ATLA_robust_RL

[2]https://github.com/umd-huang-lab/paad_adv_rl



(a) Cheetah-Run Backwards (b) Walker-Stand on One Foot

(c) Cheetah-90 Degree Push-up (d) Walker-Dance

Figure 4: Human desired behaviors behaved by the Decision Transformer under the attack of RAT.

Scenario	Task	RAT	RAT w/o h_ω	RAT w/o π_θ	RAT w/o combined policy
Manipulation	Drawer Open	**99.1%**	91.3%	21.7%	68.0%
	Drawer Close	**80.9%**	70.2%	8.0%	26.0%
Opposite	Faucet Open	84.4%	**89.8%**	0.0%	57.0%
	Faucet Close	**95.1%**	94.1%	13.0%	59.1%

Table 2: Effects of each component in RAT is evaluated based on the average attack success rate on four simulated robotic manipulation tasks. These results represent the mean success rate across five runs.

the official repositories are employed. As in most existing research (Zhang et al. 2020, 2021; Sun et al. 2022), we also use state attacks with L^∞ norm in our experiments.

Evaluation Metrics. The success metric for adversarial attacks revolves around the proximity between the task-relevant object and its final goal position, denoted as $\mathbb{I}_{\|o-t\|_2<\epsilon}$, where ϵ is a minimal distance threshold. In the manipulation scenario, we set $\epsilon = 0.05$ (5cm). For the opposite behaviors scenario, we apply the success metrics and thresholds specified for each task by Meta-world (Yu et al. 2020).

5.2 Case I: Manipulation on DRL Agents

We first conduct an evaluation of our method and other adversarial attack across two different scenarios, applying them to a range of simulated robotic manipulation tasks. Each victim agent is a well-trained SAC (Haarnoja et al. 2018) agent, specialized for a specific manipulation task and trained for 10^6 timesteps using the open-source code [3] available.

Scenarios on Manipulation. In this scenario, our objective was to manipulate the victim (robotic arm) to grasp objects at locations distant from the originally intended target, rather than completing its initial task. Table 1 presents the average attack success rates of both baseline methods and our approach across four manipulation tasks. The results indicate that the performance of RAT significantly exceeds that of the baselines by a large margin. To reduce the influence of PbRL and further highlight the advantages of RAT, we also trained baseline methods using the ground-truth reward function, labeling these as "oracle" versions. Notably, the performance of SA-RL (oracle) shows considerable improvement on several tasks compared to its preference-based counterpart. Nonetheless, RAT still outperformed SA-RL with oracle rewards in most scenarios. These findings underscore the ability of RAT to enable agents to effectively learn adversary based on human preferences. Additionally, it was observed that PA-AD struggles to perform effectively in manipulation tasks, even when trained with ground-truth rewards.

Scenarios on Opposite Behaviors. Robotic manipulation holds significant practical value in real-world applications. Therefore, we craft this scenario to quantitatively assess the vulnerability of agents proficient in various manipulation skills. In this setup, each victim agent is expected to perform the opposite of its mastered task when subjected to the manipulator's targeted attack. For instance, a victim trained to open windows would be manipulated to close them instead. As demonstrated in Table 1, RAT exhibits exceptional performance, consistently demonstrating clear advantages over the baseline methods across all tasks. This outcome reaffirms that RAT is not only effective across a broad spectrum of tasks but also capable of efficiently learning adversaries aligned with human preferences.

We observe that SA-RL and PA-AD exhibit relatively low attack success rates across numerous tasks, which can be attributed to the issue of distribution drift. This drift arises due to discrepancies between the data distribution sampled by the perturbed policy and the distribution corresponding to human-desired behaviors, leading to suboptimal performance.

5.3 Case II: Manipulation on Offline Agents

In this experiment, we show the vulnerability of offline RL agents and the capability of RAT to fool them into acting human desired behaviors. As for the implementation, we choose some online models [4] as victims, which are well-trained by official implementation with D4RL. We choose two tasks, Cheetah and Walker, using expert-level Decision Transformer agents as the victims. As illustrated in Figure 4, Decision

[3] https://github.com/denisyarats/pytorch_sac

[4] https://huggingface.co/edbeeching

Task	Model	RAT	PA-AD	SA-RL	Avg R
Door Lock	RAT-ATLA	874 ± 444	628 ± 486	503 ± 120	**668**
	RAT-WocaR	774 ± 241	527 ± 512	520 ± 236	607
	PA-ATLA	491 ± 133	483 ± 15	517 ± 129	497
	ATLA-PPO	469 ± 11	629 ± 455	583 ± 173	545
Door Unlock	RAT-ATLA	477 ± 203	745 ± 75	623 ± 60	**615**
	RAT-WocaR	525 ± 78	647 ± 502	506 ± 39	559
	PA-ATLA	398 ± 12	381 ± 11	398 ± 79	389
	ATLA-PPO	393 ± 36	377 ± 8	385 ± 26	385
Faucet Open	RAT-ATLA	442 ± 167	451 ± 96	504 ± 55	465
	RAT-WocaR	1223 ± 102	1824 ± 413	1575 ± 389	**1541**
	PA-ATLA	438 ± 53	588 ± 222	373 ± 32	466
	ATLA-PPO	610 ± 293	523 ± 137	495 ± 305	522
Faucet Close	RAT-ATLA	1048 ± 343	1223 ± 348	570 ± 453	947
	RAT-WocaR	1369 ± 158	1416 ± 208	3372 ± 1311	**2052**
	PA-ATLA	661 ± 279	371 ± 65	704 ± 239	538
	ATLA-PPO	1362 ± 149	688 ± 196	426 ± 120	825

Table 3: Average return ± standard deviation of robust agents under various attacks, averaged over 100 episodes.

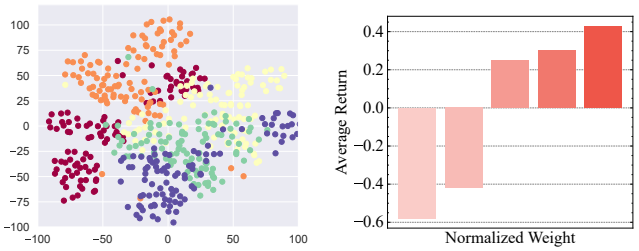
Transformer reveals weaknesses that can be exploited, leading it to execute human-preferred behaviors rather than its intended tasks. Under adversarial manipulation, the Cheetah agent is shown to run backwards rapidly in Figure 4a and perform a 90-degree push-up in Figure 4c. Meanwhile, the Walker agent maintains superior balance on one foot in Figure 4b and appears to dance with one leg raised in Figure 4d. These outcomes indicate that RAT is effective in manipulating these victim agents towards behaviors consistent with human preferences, highlighting the significant vulnerability of embodied agents to strong adversaries.

5.4 Robust Agents Training and Evaluation

A practical application of RAT is in assessing the robustness of established models or in enhancing an agent's robustness. ATLA-PPO (Zhang et al. 2021) presents a generic training framework aimed at improving robustness, which involves alternating training between an agent and an SA-RL attacker. PA-ATLA (Sun et al. 2022) follows a similar approach but employs a more advanced RL attacker, PA-AD. Drawing inspiration from previous works (Zhang et al. 2021; Liang et al. 2022b), we introduce two novel robust training methods: RAT-ATLA and RAT-WocaR. RAT-ATLA's central strategy is to alternately train an agent and a RAT attacker, whereas RAT-WocaR focuses on directly estimating and minimizing the reward of the intention policy, obviating the need for extra samples to learn an attacker. Table 3 compares the effectiveness of RAT-ATLA and RAT-WocaR for SAC agents on robotic simulation manipulation tasks against leading robust training methods. The experimental findings highlight two key points: first, RAT-ATLA and RAT-WocaR substantially improve agent robustness; and second, RAT is capable of executing stronger attacks on robust agents, showcasing its effectiveness in challenging environments.

5.5 Ablation Studies

Contribution of Each Component. We further investigate the effect of each component in RAT. RAT involves three essential components or techniques: the intention policy π_θ,



(a) t-SNE Visualization (b) Weight Visualization

Figure 5: **Effects of the Weighting Function.** (a) Trajectory weights generated by the weighting function from various policies are visualized with t-SNE. (b) A visualization of the weights of trajectories of different qualities by five policies.

the weighting function h_ω and the combined behavior policy. Table 2 shows π_θ emerges as a pivotal component in RAT, significantly boosting the attack success rate. This enhancement is largely due to its capability to mitigate distribution drift between the victim's behavior and the desired behavior.

Effects of the Weighting Function. To further understand the weighting function proposed in Section 4.2, we conduct comprehensive experimental data analysis and visualization from multiple perspectives. We sample five perturbed policies uniformly, each representing a progressive stage of performance improvement before the convergence of RAT. For each of these policies, 100 trajectories were rolled out, and their corresponding trajectory weight vectors were obtained via the weighting function. Utilizing t-SNE (van der Maaten and Hinton 2008) for visualization, Figure 5a showcases the weight vectors of different policies. This illustration reveals distinct boundaries between the trajectory weights of various policies, indicating the weighting function's ability to differentiate trajectories based on their quality. In Figure 5b, trajectories with higher success rates in manipulation are represented in darker colors. The visualization suggests that the weighting function assigns higher weights to more successful trajectories, thereby facilitating the improvement of the adversary's performance.

6 Conclusion

In this paper, we propose RAT, a targeted behavior attack approach against DRL learners, which manipulates the victim to perform human-desired behaviors. RAT involves an adversary adding imperceptible perturbations on the observations of the victim, an intention policy as a flexible behavior target, and a weighting function to identify essential states for the efficient adversarial attack. We analyze the convergence of RAT and prove that RAT converges to critical points under some mild conditions. Empirically, we design two scenarios on several manipulation tasks in Meta-world, and the results demonstrate that RAT outperforms the baselines in the targeted adversarial setting. Additionally, RAT can enhance the robustness of agents via adversarial training. We further show embodied agents' vulnerability by attacking Decision Transformer on some MuJoCo tasks.

References

Bai, F.; Zhang, H.; Tao, T.; Wu, Z.; Wang, Y.; and Xu, B. 2023. PiCor: Multi-Task Deep Reinforcement Learning with Policy Correction. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 37(6): 6728–6736.

Bai, F.; Zhao, R.; Zhang, H.; Cui, S.; Wen, Y.; Yang, Y.; Xu, B.; and Han, L. 2024. Efficient Preference-based Reinforcement Learning via Aligned Experience Estimation. *arXiv preprint arXiv:2405.18688*.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4): 324–345.

Buddareddygari, P.; Zhang, T.; Yang, Y.; and Ren, Y. 2022. Targeted Attack on Deep RL-based Autonomous Driving with Learned Visual Patterns. In *International Conference on Robotics and Automation (ICRA)*, 10571–10577.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 15084–15097. Curran Associates, Inc.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc.

Fischer, M.; Mirman, M.; Stalder, S.; and Vechev, M. 2019. Online robustness training for deep reinforcement learning. *arXiv preprint arXiv:1911.00887*.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning (ICML)*, volume 80, 1861–1870.

Hasanbeig, M.; Kroening, D.; and Abate, A. 2020. Deep Reinforcement Learning with Temporal Logics. In *Formal Modeling and Analysis of Timed Systems (FORMATS)*, 1–22.

Huang, S. H.; Papernot, N.; Goodfellow, I. J.; Duan, Y.; and Abbeel, P. 2017. Adversarial Attacks on Neural Network Policies. In *International Conference on Learning Representations (ICLR)*.

Hussenot, L.; Geist, M.; and Pietquin, O. 2019a. Targeted Attacks on Deep Reinforcement Learning Agents through Adversarial Observations. abs/1905.12282.

Hussenot, L.; Geist, M.; and Pietquin, O. 2019b. Targeted Attacks on Deep Reinforcement Learning Agents through Adversarial Observations. *CoRR*, abs/1905.12282.

Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; and Amodei, D. 2018. Reward learning from human preferences and demonstrations in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc.

Jia, X.; Yang, Z.; Li, Q.; Zhang, Z.; and Yan, J. 2024. Bench2Drive: Towards Multi-Ability Benchmarking of Closed-Loop End-To-End Autonomous Driving. *arXiv preprint arXiv:2406.03877*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*, 521(7553): 436–444.

Lee, K.; Smith, L. M.; and Abbeel, P. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *International Conference on Machine Learning (ICML)*, volume 139, 6152–6163.

Liang, X.; Shu, K.; Lee, K.; and Abbeel, P. 2022a. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

Liang, Y.; Sun, Y.; Zheng, R.; and Huang, F. 2022b. Efficient Adversarial Training without Attacking: Worst-Case-Aware Robust Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 22547–22561.

Lin, Y.-C.; Hong, Z.-W.; Liao, Y.-H.; Shih, M.-L.; Liu, M.-Y.; and Sun, M. 2017a. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3756–3762.

Lin, Y.-C.; Hong, Z.-W.; Liao, Y.-H.; Shih, M.-L.; Liu, M.-Y.; and Sun, M. 2017b. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. In *IJCAI*, 3756–3762.

Liu, R.; Bai, F.; Du, Y.; and Yang, Y. 2022. Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, R.; Du, Y.; Bai, F.; Lyu, J.; and Li, X. 2024. PEARL: Zero-shot Cross-task Preference Alignment and Robust Reward Learning for Robotic Manipulation. In *International Conference on Machine Learning (ICML)*.

Lütjens, B.; Everett, M.; and How, J. P. 2020. Certified Adversarial Robustness for Deep Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, volume 100, 1328–1337.

Mandlekar, A.; Zhu, Y.; Garg, A.; Fei-Fei, L.; and Savarese, S. 2017. Adversarially Robust Policy Learning: Active construction of physically-plausible perturbations. In *International Conference on Intelligent Robots and Systems (IROS)*, 3932–3939.

Oikarinen, T.; Zhang, W.; Megretski, A.; Daniel, L.; and Weng, T.-W. 2021. Robust Deep Reinforcement Learning through Adversarial Loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 26156–26167.

Park, J.; Seo, Y.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2022. SURF: Semi-supervised Reward Learning with Data

Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust Adversarial Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, 2817–2826. PMLR.

Shen, Q.; Li, Y.; Jiang, H.; Wang, Z.; and Zhao, T. 2020. Deep Reinforcement Learning with Robust and Smooth Policy. In *International Conference on Machine Learning (ICML)*, volume 119, 8707–8718.

Shi, Y.; Wen, M.; Zhang, Q.; Zhang, W.; Liu, C.; and Liu, W. 2024. Autonomous Goal Detection and Cessation in Reinforcement Learning: A Case Study on Source Term Estimation. *arXiv preprint arXiv:2409.09541*.

Sun, Y.; Zheng, R.; Liang, Y.; and Huang, F. 2022. Who Is the Strongest Enemy? Towards Optimal and Efficient Evasion Attacks in Deep RL. In *International Conference on Learning Representations (ICLR)*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems (IROS)*, 5026–5033.

Vamplew, P.; Smith, B. J.; Källström, J.; Ramos, G.; Rădulescu, R.; Roijers, D. M.; Hayes, C. F.; Heintz, F.; Mannion, P.; Libin, P. J.; et al. 2022. Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 36(2): 41.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Wang, X.; Tian, Z.; Wan, Z.; Wen, Y.; Wang, J.; and Zhang, W. 2023. Order Matters: Agent-by-agent Policy Optimization. *11th ICLR*.

Wang*, X.; Zhang*, S.; Zhang, W.; Dong, W.; Chen, J.; Wen, Y.; and Zhang, W. 2024. ZSC-Eval: An Evaluation Toolkit and Benchmark for Multi-agent Zero-shot Coordination. *Advances in neural information processing systems (NeurIPS) Track on Datasets and Benchmarks*.

Wen, M.; Wan, Z.; Wang, J.; Zhang, W.; and Wen, Y. 2024. Reinforcing LLM Agents via Policy Optimization with Action Decomposition. In *Advances in neural information processing systems (NeurIPS)*.

Xu, Y.; and Yu, L. 2023. DRL-Based Trajectory Tracking for Motion-Related Modules in Autonomous Driving. *arXiv preprint arXiv:2308.15991*.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, 1094–1100. PMLR.

Zhang, H.; Bai, F.; Xiao, C.; Gao, C.; Xu, B.; and Müller, M. 2025. β-DQN: Improving Deep Q-Learning By Evolving the Behavior. arXiv:2501.00913.

Zhang, H.; Chen, H.; Boning, D. S.; and Hsieh, C.-J. 2021. Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. In *International Conference on Learning Representations (ICLR)*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D.; and Hsieh, C.-J. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 21024–21037. Curran Associates, Inc.

Zhang, H.; Sun, K.; bo xu; Kong, L.; and Müller, M. 2024a. A Distance-based Anomaly Detection Framework for Deep Reinforcement Learning. *Transactions on Machine Learning Research*.

Zhang, H.; Xiao, C.; Gao, C.; Wang, H.; bo xu; and Müller, M. 2024b. Exploiting the Replay Memory Before Exploring the Environment: Enhancing Reinforcement Learning Through Empirical MDP Iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhang, H.; Xiao, C.; Wang, H.; Jin, J.; bo xu; and Müller, M. 2023. Replay Memory as An Empirical MDP: Combining Conservative Estimation with Experience Replay. In *International Conference on Learning Representations (ICLR)*.