AI Open 5 (2024) 126–141

KeAi
CHINESE ROOTS
GLOBAL IMPACT

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

Full length article

A study of natural robustness of deep reinforcement learning algorithms towards adversarial perturbations

Qisai Liu [a], Xian Yeow Lee [a], Soumik Sarkar [a,b,*]

[a] *Department of Mechanical Engineering, Iowa State University, Ames, 50011, IA, United States*
[b] *Department of Computer Science, Iowa State University, Ames, 50011, IA, United States*

ARTICLE INFO

Dataset link: https://github.com/super864/Natural-Robustness-RL

Keywords:
Deep reinforcement learning
Adversarial attacks
Black-box attacks
White-box attack
Heuristic-based attacks
Gradient-based attacks
Robustness

ABSTRACT

Deep reinforcement learning (DRL) has been shown to have numerous potential applications in the real world. However, DRL algorithms are still extremely sensitive to noise and adversarial perturbations, hence inhibiting the deployment of RL in many real-life applications. Analyzing the robustness of DRL algorithms to adversarial attacks is an important prerequisite to enabling the widespread adoption of DRL algorithms. Common perturbations on DRL frameworks during test time include perturbations to the observation and the action channel. Compared with observation channel attacks, action channel attacks are less studied; hence, few comparisons exist that compare the effectiveness of these attacks in DRL literature. In this work, we examined the effectiveness of these two paradigms of attacks on common DRL algorithms and studied the natural robustness of DRL algorithms towards various adversarial attacks in hopes of gaining insights into the individual response of each type of algorithm under different attack conditions.

1. Introduction

Deep reinforcement learning (DRL) has seen substantial successes in multiple domains of applications such as design (Abbeel et al., 2006), scheduling (Wang et al., 2019) and robotic control applications in industrial automation (Bahrin et al., 2016). Contrary to supervised learning, RL algorithms train an agent to learn to perform a given task in an environment by making sequential actions and observing the resulting rewards to learn an optimal policy. In recent years, advancements in neural networks have led to the popularity of DRL, where a deep neural network represents the RL policy. Although neural networks are powerful function approximators, they are also extremely easy to fool into making erroneous predictions by applying perturbation on the model's inputs (Goodfellow et al., 2014). This observation led to numerous studies on the robustness of deep learning algorithms. A study by Huang et al. (2017) proved that similar adversarial attacks could also be extended to manipulate RL agents where the RL agent is vulnerable to subtle adversarial attacks that are not perceivable to humans but could cause a significant change in RL policy's actions. Subsequently, this has led to the development of several other successful adversarial attacks (Behzadan and Munir, 2017; Lin et al., 2017; Pattanaik et al., 2017; Xiao et al., 2019; Lee et al., 2020).

While numerous works have developed DRL algorithms that are robust towards different perturbations (Tan et al., 2020; Zhang et al., 2020, 2021; Moos et al., 2022), to the best of our knowledge, a study that compares the response of popular benchmark DRL algorithms towards common adversarial perturbations is still lacking in the literature, and this work aims to fill in such a gap. Specifically, in this work, we analyze the performance of multiple DRL algorithms commonly used in literature when subjected to observation and action perturbations. Most of the perturbations we study in this work are also commonly used perturbation methods in the literature. Although these perturbations are typically generated using different methods, in practice they could manifest in many forms, such as a result of malicious attacks aimed at degrading the RL algorithm's performance or as external environmental factors such as faulty sensors or sudden system load that an RL-based controller needs to handle (Jan et al., 2021; Argawal et al., 2021). As a first step, we restrict our experiments to existing model-free RL algorithms that are designed for continuous action space environments, which provide a more realistic proxy to industrial robotic applications, where adversarial attacks are of greater concern. Our experiments aim to answer the following questions: **(1)** Are the existing DRL algorithms especially sensitive to one class of adversarial perturbations over the other? (e.g., observation vs. action space perturbations), **(2)** Is there a specific DRL algorithm that is naturally more robust than all other algorithms under adversarial perturbations and **(3)** Is there a limitation of the magnitude of the perturbation on the degradation of the DRL performance, i.e., is there

[*] Corresponding author at: Department of Mechanical Engineering, Iowa State University, Ames, 50011, IA, United States.
E-mail address: soumiks@iastate.edu (S. Sarkar).

https://doi.org/10.1016/j.aiopen.2024.08.005
Received 6 June 2023; Received in revised form 13 August 2024; Accepted 26 August 2024
Available online 3 September 2024

an empirically observable threshold in which perturbations below or above this threshold will not affect the behavior of the DRL policy? As such, we aim to assess the performance of agents across perturbation levels and identify the range of perturbation that has the greatest impact on each agent, rather than the effect of the magnitudes on the imperceptibility of the perturbations. Additionally, we intend to rank the algorithms based on their ability to perform under varying levels of difficulty and different attack strategies. The results of our experiments are intended to offer insights into the robustness of these DRL algorithms and serve as a stepping stone for the development of more robust DRL algorithms in the future.

2. Related works

2.1. Introduction to adversarial attack

Adversarial attacks on deep neural networks were first popularized by Szegedy et al. (2013), who demonstrated that small perturbations added to input images could lead to significant misclassifications by image classification models. These attacks expose vulnerabilities in DNNs, challenging their robustness and reliability.

2.2. Types of adversarial attack

Adversarial attacks are generally categorized based on the adversary's knowledge of the machine learning (ML) model:

1. **white-box attacks:** The adversary has full access to the model's internal parameters, such as the learned weights, training parameters, and training and testing data. With that complete information, these attacks can be used to analyze more precise manipulation, often used to evaluate the worst-case scenarios (Ebrahimi et al., 2017).
2. **Gray-box attacks:** The adversary possesses partial knowledge of the model. This type of attack falls between the extremes of white-box and black-box, leveraging some known aspects while remaining uncertain about others (Papernot et al., 2016a; Tramèr et al., 2017; Carlini et al., 2019; Papernot et al., 2017).
3. **black-box attacks:** The adversary has no knowledge of the model's internal workings and instead relies on external observations, such as inputs and outputs, to infer vulnerabilities. For example, the author of Mahmood et al. (2021) provides a novel analysis to comprehend the success rate of attacks with respect to each adversarial model. Hence, black-box attacks are often established based on the model inputs, confidence scores, or perturbing the feedback of the ML model (Guo et al., 2019).

2.3. Adversarial attacks on deep reinforcement learning

In the realm of DRL, adversarial attacks have also been shown to be effective. Behzadan and Munir (2017) highlighted that Deep Q-Networks (DQN) are vulnerable to adversarial state perturbations. The adversarial perturbations were generated using the Fast Gradient Sign Method (FGSM) and Jacobian-based Saliency Map Attack (JSMA) (Papernot et al., 2016b). Additionally, they also implemented a black-box and showed a success rate of 70%.

Expanding on this, in Huang et al. (2017), the authors employed similar attack techniques in Behzadan and Munir (2017) but implemented the attacks on other DRL algorithms such as DQN, Trust Region Policy Optimization (TRPO), and Asynchronous Advantage Actor-Critic (A3C) methods in both white and black-box settings. Their results demonstrated that DQN was particularly susceptible to adversarial manipulations, suggesting a need for more robust defense mechanisms.

Recent studies have further explored the impact of adversarial attacks in multi-agent reinforcement learning systems. Gleave et al. (2020) showed that adversarial perturbations could be introduced by a compromised agent, leading to manipulated behaviors in other agents within the system. The development of more sophisticated black-box attack methods continues to be a critical area of research. Another work suggests various enhancements to the black-box adversarial attack method known as SimBA, with the goal of improving its efficiency by optimizing query usage (Yang et al., 2020).

Nonetheless, in this work, we will limit the scope of our experiments and the class of perturbations that are only applicable to single-agent environments. In addition to observation space perturbations, the attacks can also occur in the action space in the form of perturbations on the actuators. For example, Lee et al. (2020) proposed spatial–temporal coupled action space attacks that reveal the potential vulnerabilities of the DRL model. Moreover, action space perturbations can also manifest in the form of environmental noise or changes in environmental factors (Tan et al., 2020). In addition, changes in environmental factors may also manifest as a form of environmental perturbation that affects the underlying dynamics of the system (Sun et al., 2022).

2.4. Recent innovations

For brevity, we refer interested readers to the more detailed and complete taxonomy of adversarial attacks presented in Chen et al. (2019). In Li et al. (2023), the authors introduce an innovative adversarial strategy that integrates Attack Time Selection and Optimal Attack Action to precisely target DRL systems by exploiting vulnerabilities at critical decision points, showcasing a methodical approach to enhancing attack efficacy and stealth.

The authors of Li et al. (2022) offer a comprehensive overview of adversarial attacks on DRL systems. They emphasize the strategic deployment of attacks across observation, reward, action, policy, and environment vectors to degrade system performance, additionally, the paper also discusses the development of defensive strategies aimed at increasing the robustness of DRL systems against such attacks.

2.5. Summary

This overview highlights the evolution of adversarial attacks from their origins in DNNs to their application in DRL systems. While substantial progress has been made in understanding these attacks and developing countermeasures, challenges remain in ensuring the robustness and security of DRL models. Our work builds on this foundation by focusing on action space attacks, with the aim of deepening our understanding of their analysis and performance across different white-box and black-box scenarios.

3. Methodology

In this section, we provide a description of the experiments we conducted to compare the performance and response of common DRL algorithms to adversarial attacks.

3.1. Selection of environments and algorithms

We conducted our experiments on five different continuous control environments based on OpenAI Gym (Brockman et al., 2016) MuJoCo environments. The five selected environments are: (i) Ant, (ii) HalfCheetah, (iii) Swimmer, (iv) Walker, and (v) Hopper. To facilitate a more accurate comparison, all experiments were run with six random seeds, and for each seed, we ran 100 episodes and reported the average score across all episodes and seeds. These five MuJoCo environments were selected due to the fact that they are well-established benchmarks in the RL community and have been shown to be solvable by popular RL algorithms without requiring additional reward engineering and with a feasible amount of computation. This makes them an excellent choice for comparing the effectiveness of different algorithms. Moreover, these environments also simulate various action and observation spaces with different dimensionality and difficulty, thus closely mimicking real-world scenarios and the challenges that RL agents face under perturbations. To compare the performance of various DRL algorithms in continuous control tasks, we chose five widely-used algorithms that are commonly adopted as benchmarks. These include Proximal Policy Optimization (PPO) (Schulman et al., 2017), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), Twin Delayed DDPG (TD3) (Fujimoto et al., 2018), and Soft Actor-Critic (SAC) (Haarnoja et al., 2018). Despite the emergence of newer RL algorithms, we chose to study these five algorithms because they are still prevalent in research and practical applications due to their widespread adoption as benchmarks and stability during training.

3.2. Black-box attacks

Next, we describe the suite of black-box attacks that we implemented as part of our experiments to compare the performance of the DRL agents. To the best of our knowledge, there are no black-box attacks that are consistently used as a benchmark due to the fact that most black-box attacks typically require querying the model and different methods have made different assumptions on the amount of information querying a model might reveal. As such, as an initial step, we limit the scope of black-box attacks in this paper to a set of heuristics that consist of simple additive perturbations. In practice, these black-box perturbations may represent environmental factors such as additional friction or load acting on a robot in realistic conditions. However, we highlight that black-box attack strategies extend beyond naive additive perturbations and will be a key focus of future studies. To fully investigate the behavior of the policies in a comprehensive manner, we develop multiple heuristic strategies for black-box attacks. These strategies were generated by identifying the three stages where the perturbations can be performed. The first stage consists of the channel of perturbation, where the perturbation can either be added to the observations of the agent or the actions of the agent. The second stage involves the magnitude of perturbation, where the magnitude of perturbation is either random, bounded by the action space, or bounded by the magnitude of the actual action taken by the agent. The third stage involves the direction in which the perturbation is applied. Since the actions and observations in these environments are multi-dimensional vectors, the perturbations can be added in four different ways: (1) consistently adding noise following the signs of individual observations/actions, (2) consistently adding noise that is opposite the sign of the individual observations/actions, (3) Chooses a perturbation direction randomly at each time step, applying it uniformly across all dimensions of observations or actions within that timeframe, ensuring consistent perturbation direction for every observation or action. and (4) This strategy selects a distinct perturbation direction for each observation or action at every time step. This introduces a higher level of unpredictability and granularity, with each observation or action potentially experiencing different directional perturbations within the same time frame, leading to a varied and more disruptive perturbation pattern. The suite of all possible black-box attacks can be summarized according to Fig. 1, where selecting a choice at each stage will result in a valid strategy.

3.3. White-box attacks

Next, we describe the three white-box attack strategies that we selected to test on the common benchmark RL algorithms. All three attacks leverage the gradient information to craft the attacks and these three attack strategies were selected to study both perturbations on the observation and action channels. Specifically, the white-box attacks we implemented are the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014), Projected Gradient Descent (PGD) algorithm (Madry et al., 2017), and the Myopic Action Space (MAS) attack algorithm (Lee et al., 2020).

Fast Gradient Sign Method: FGSM generates a perturbation by taking the sign of the loss's gradient with respect to observation and adding that perturbation to the observation. By adding the perturbation, FGSM seeks to find a perturbation that increases the loss function, hence fooling the agent into taking poor action. Formally, the FGSM method can be defined as follows:

$$\hat{s} = s + \epsilon \times sign\nabla_s(L) \tag{1}$$

where s and $\hat{s}$ denote the original and perturbed observation respectively, ϵ denotes a budget that scales the perturbation to keep it undetectable, and L denotes the policy's loss function. To instantiate these attacks in practice, we used the actor network's loss as the loss function to obtain the gradients to compute the perturbation for each of the RL benchmark algorithms.

Projected Gradient Descent: PGD is an iterative attack method that works similarly to FGSM in principle. While the FGSM attacks only take a single gradient step, the PGD performs multiple gradient steps to maximize the loss function and finally projects the perturbation back into the budget of ϵ. In our implementation, we set the number of iterations of PGD to be 25 after empirically observing that the degradation in performance of the RL policy displays no significant changes after 25 iterations.

Myopic Action Space attack algorithm: MAS is an attack algorithm that generates perturbation that attacks the action channel rather than the observation channel. It follows the same principle of FGSM and PGD of generating perturbations but takes gradients of the reward function with respect to the action instead of the observation. Since the gradients of the reward function with respect to the action might not be accessible, the gradients of the action probabilities or value function are taken as a proxy of the reward function to generate the perturbation, which is subsequently added to the RL agent's actions.

4. Results and discussions

In this section, we present the results of our experiments comparing the performance of the RL algorithms when subjected to the different adversarial attacks as discussed in the previous section. To fully understand the efficacy of each attack, we first trained the five RL policies using PFRL's implementation (Fujita et al., 2021) on the five MuJoCo environments and ensured that the final rewards are similar to the reported scores. As such, the subsequent results are all based on mounting the attacks on the trained RL policies during test time. All experiments were performed on an internal cluster using three GeForce GTX TITAN X GPUs for training the RL agents and Intel(R) Xeon(R) CPU E5-1650 v3 CPUs for testing and mounting the attacks.

4.1. Comparison of different black-box attack strategies

We begin by visualizing and comparing the effects of different attack strategies on the five RL algorithms. To measure the effectiveness of each attack, we measure the percentage change in rewards, denoted as $\Delta R\%$, and defined as the change in rewards due to an attack as a fraction of the original rewards achieved by the trained policy. To compare the attacks, we plot the $\Delta R\%$ for each RL algorithm as a stacked bar plot to measure the overall effectiveness of each attack strategy on all the algorithms. As an illustrative example, we show the comparison for the HalfCheetah environment in Fig. 2. An important parameter when mounting these attacks is the constraint on the magnitude of the perturbations or the attack budget, ϵ. To obtain a comprehensive comparison, we mounted all the attacks at four budget levels: 25%, 50%, 100%, and 200%.

As shown in Fig. 2, the first observation that can be made is that all the different attack strategies resulted in a negative $\Delta R\%$ across all RL

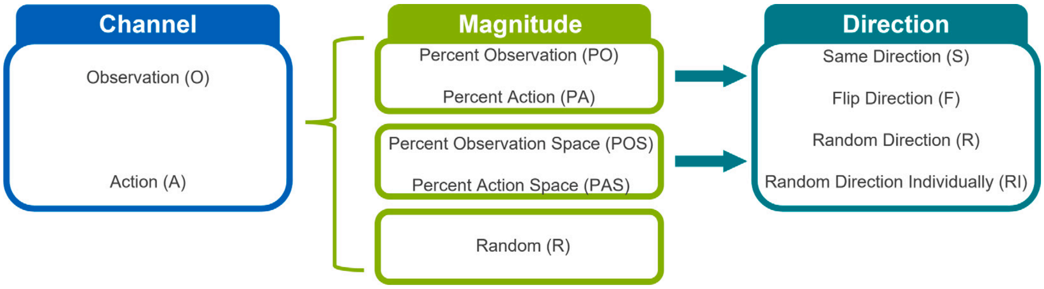


Fig. 1. Black-box attack strategies: The flowchart shows the various black-box strategies implemented. The attacks can be mounted on either one of the two channels, with the constraint on the attack following one of the three magnitudes and the specific instantiation following one of the four directions. The names of each attack are denoted by the abbreviation from each stage, e,g: attacks on the observation with constraint from percent of observation with the same direction denotes O_PO_S.

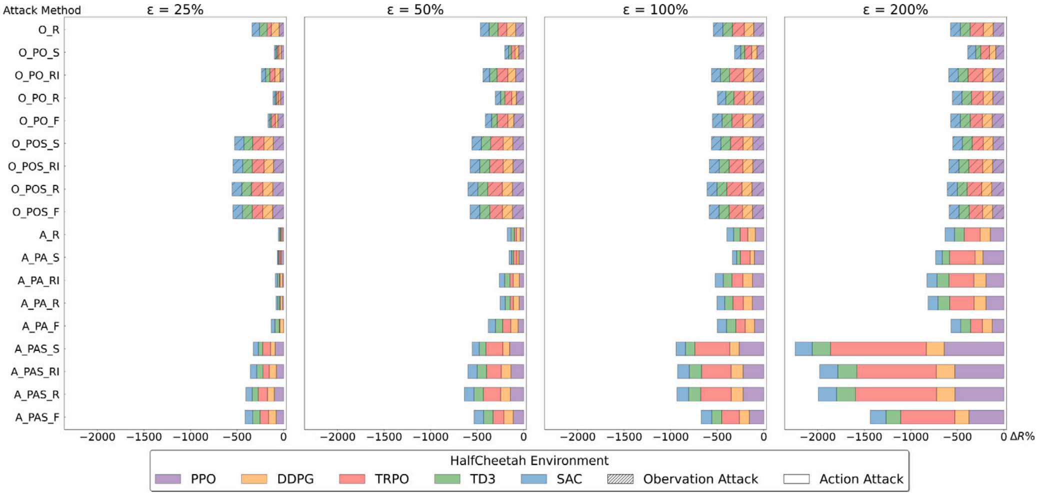


Fig. 2. Comparison of black-box attack on HalfCheetah: Vertical axis represents different black-box attack strategies, and the horizontal axis denotes the cumulative $\Delta R\%$ across all RL algorithms. The colors represent different RL algorithms, bars with shaded patterns represent observation channel attacks, and solid bars represent action channel attacks. Each subplot denotes mounting the attacks with a specific budget ϵ on the magnitude of the perturbations.

algorithms in the HalfCheetah environments across all ϵ values. (more detailed framework is represented in Fig. A.8 in Supplementary)

We did, however, observe certain environments seen in supplementary Figs. A.8, A.10, A.9, and A.11 where some attack strategies resulted in a positive $\Delta R\%$; nonetheless, the overall trend remains negative. For more detailed visualization plots with normalization, please refer to the appendix Fig. A.16, A.17, A.18, A.19, A.20. The second observation we made is that even the most ineffective attack strategies saturate when the budget is above 100%, with the most drastic changes in $\Delta R\%$ occurring below the budget of 100%. As such, in our following experiments below, we focused only on budget levels below 100% but at a finer resolution.

Comparing the attack strategies on the observation channel versus the action channels, we observed that overall, attacks on the observation channel are as effective and sometimes more effective (in Hopper, Swimmer, and Walker) than action channel attacks up to a certain budget value, specifically for $\epsilon = 25\%$ and 50%. Beyond $\epsilon = 50\%$, the action channel attacks are more effective while the observation channel generally saturates (elaborated in further detail in the following sections). In terms of the different strategies of attacks, overall, we observed that strategies that add a perturbation that has an opposite sign (flip direction) to the original action/observation (O_PO_F, O_POS_F, A_PA_F, A_PAS_F) value are the most effective, while strategies that add a perturbation that has the same sign (same direction) (A_PA_S, A_PAS_S, O_PO_S, O_POS_S) are the least effective. Furthermore, the attack strategy that perturbs the individual elements of the observation channel/action channel (O_PO_RI, O_POS_RI, A_PA_RI, A_PAS_RI) is also slightly more effective than randomly perturbing the entire vector in a random direction (O_R, A_R). We highlight that these trends are observed consistently across all five benchmark algorithms and all environments, except for the Ant environment.

Based on our observations for the Ant environment (as shown in Appendix Fig A.8), the response of the benchmark algorithm towards action channel attacks deviates slightly from the rest of the environments. Specifically, the attacks that add perturbation in an opposing direction in the action space (A_PA_F, A_PAS_F) resulted in a positive $\Delta R\%$ for most RL algorithms, while perturbations of the same direction (A_PAS_S) ended up being one of the most effective strategies. We hypothesize that this is possibly due to the more complex 3-dimensional non-linear interaction of the action space of the Ant robot as compared to the rest of the environments, which are restricted to the 2-dimensional plane. This also alludes to the fact that the benchmark RL algorithms have not converged to the optimal policies in practice and the perturbations ended up being less effective. Nonetheless, the Ant environment also poses a challenge for comparison with other environments due to its extremely large state space with mostly zero values. This differs from other environments where the state spaces are dense vectors that can be fully utilized by the RL agent. Therefore, to fully comprehend the behavior of agents in such complex environments and to further validate our hypothesis, additional experimentation, and analysis are necessary in future works.

4.2. Comparison of the robustness of different policies

Next, we present the comparisons between the overall robustness of different RL policies across all environments and black-box attacks. The sum of all the $\Delta R\%$ across all attacks and all environments for each policy is illustrated in Fig. 3. We hope that such a plot would reveal

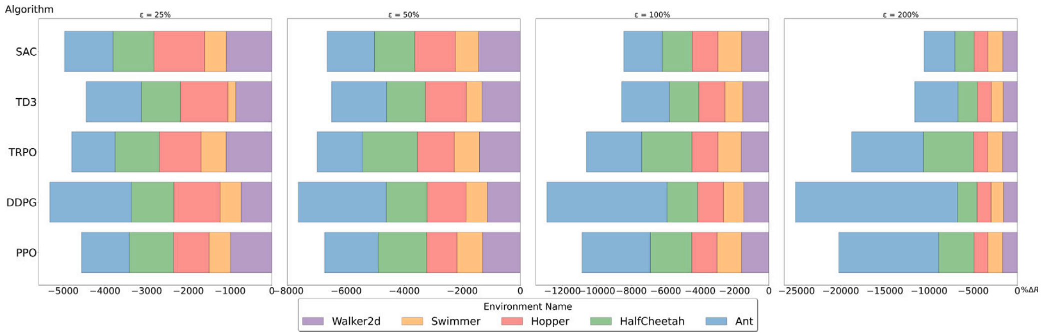


Fig. 3. Comparison of the robustness of different policies: The vertical axis represents different RL algorithms, and the horizontal axis denotes the cumulative $\Delta R\%$ across all black-box attacks. Different colors denote different environments and each subplot represents mounting the attacks with a specific budget ϵ on the magnitude of the perturbations, in the order of 25%, 50%, 100%, and 200%.

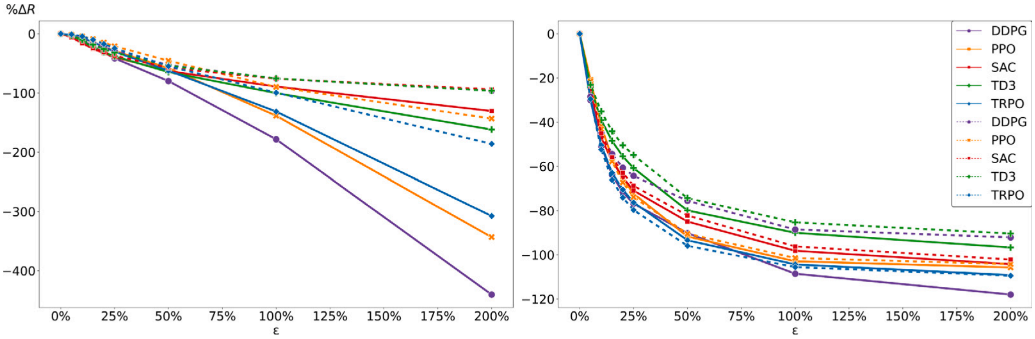


Fig. 4. Effect of increasing ϵ on $\Delta R\%$ in HalfCheetah: The x-axis represents ϵ and the y-axis represents the $\Delta R\%$ with respect to ϵ. The solid line represents the average $\Delta R\%$ across all black box attacks and environments, and the dotted line represents the average excluding the Ant environment.

the natural robustness of each type of policy, i.e., how insensitive an RL algorithm is to perturbations if it was not specifically trained to be robust in the first place. In summary, we observed that both TD3 and SAC exhibit the most robustness across all environments and for all values of budget on the magnitude of perturbation, ϵ. Additionally, TD3 and SAC were also the least affected when increasing ϵ, while the other three algorithms $\Delta R\%$ increased significantly, with the Ant and HalfCheetah environments contributing to most of the change. On the other hand, we note that DDPG was overall the policy that is most sensitive to perturbations, especially in the Ant environment. Removing the outlier effect of the Ant environment, TRPO ranks as the policy most vulnerable to perturbations. While it is not clear why DDPG or TRPO are so sensitive, we hypothesize that the reason both SAC and TD3 are more robust is because of their shared implementation of having two Q-values to reduce overestimation. More importantly, SAC also includes an entropy bonus term in the objective function, while TD3 implements a target policy smoothing that includes noise to the action, both of which can be considered an indirect way of incorporating adversarial training in the learning process. However, this hypothesis remains to be further verified.

4.3. The effect of budget ϵ on $\Delta R\%$

While it is clear that the value of ϵ, the budget on the magnitude of perturbation, affects the effectiveness of an attack in a positively correlated manner, we study the relationship between these two variables in more detail in this section. We repeated the experiments shown in Fig. 2 by varying the values of ϵ at a finer resolution of 5%. The solid lines in Figs. 4(a) and (b) represent the average of $\Delta R\%$ across all environments for each RL algorithm for action channel attacks and observation channel attacks, respectively. From this plot, we can make several more interesting observations. Firstly, we see that perturbations in the observation channel are effective but have diminishing effectiveness, as seen by the saturating trends of the $\Delta R\%$ in Fig. 4(b). In contrast, perturbations in the action space do not display this characteristic as we see that $\Delta R\%$ still decreases at a linear rate as the attack budget increases up until 200%. However, one caveat is that a major contributor to the continued decrease in rewards was due to the attacks mounted on the Ant environment, as discussed in the previous section. Removing the Ant environment from the trends (as shown in the dotted lines of Figs. 4(a) and (b)) revealed that the $\Delta R\%$ decreases less drastically for action channel attacks but still more significant than observation channel attacks. This further validates our hypothesis that RL agents that operate in environments with a higher degree of freedom are likely to be more sensitive to perturbations and display catastrophic failures.

Another observation that can be made is that in the regime of the $\epsilon \leq 50\%$, attacks on the agent's observation channel cause a much more significant drop in performance than attacks on the agent's action channel. This observation is further validated when we visualize the $\Delta R\%$ for every 5% increment of ϵ in Fig. 5. Once again, we only present the experiments for HalfCheetah for brevity, with the visualization for the rest of the environments shown in supplementary Figs. A.12, A.13, A.14, and A.15. For more detailed visualization plots with normalization, please refer to the appendix Figs. A.20, A.21, A.22, A.23, A.24, A.25. From the figure, we can observe that the largest drop in rewards for observation channel attacks (bars with diagonal patterns) occurs when ϵ is between 0 to 10%. Meanwhile, we observe the exact opposite trend in action channel attacks where the initial effect when ϵ is between 0 to 5% was small, but the $\Delta R\%$ increases as we increase ϵ.

4.4. Comparison of different white-box attacks

Next, we compare the effects of the three white-box attack strategies we had selected on the performance of the benchmark RL policies. While this is by no means a comprehensive experiment of white-box attack strategies, we hope that the results in these sections will

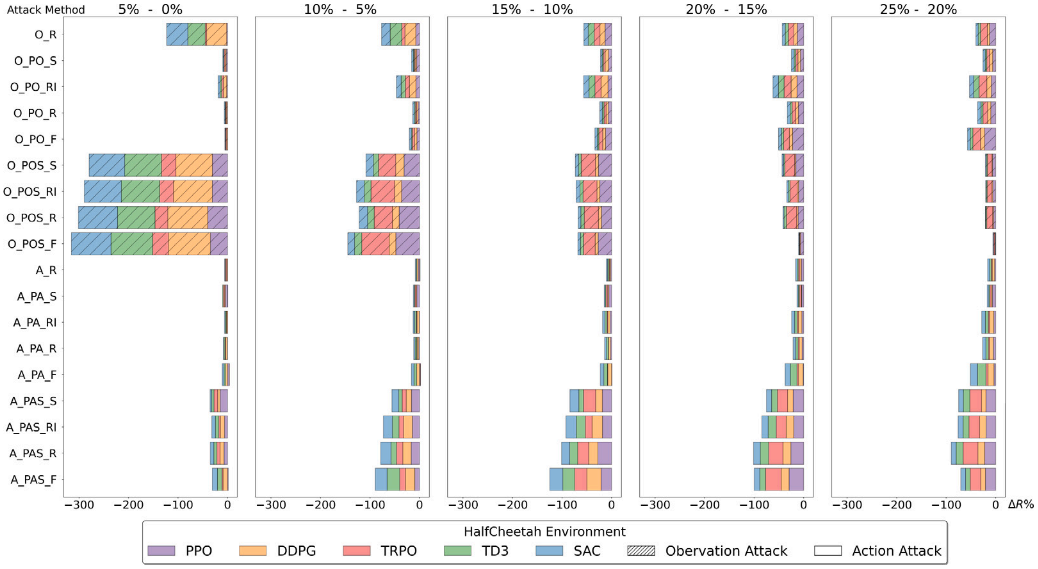
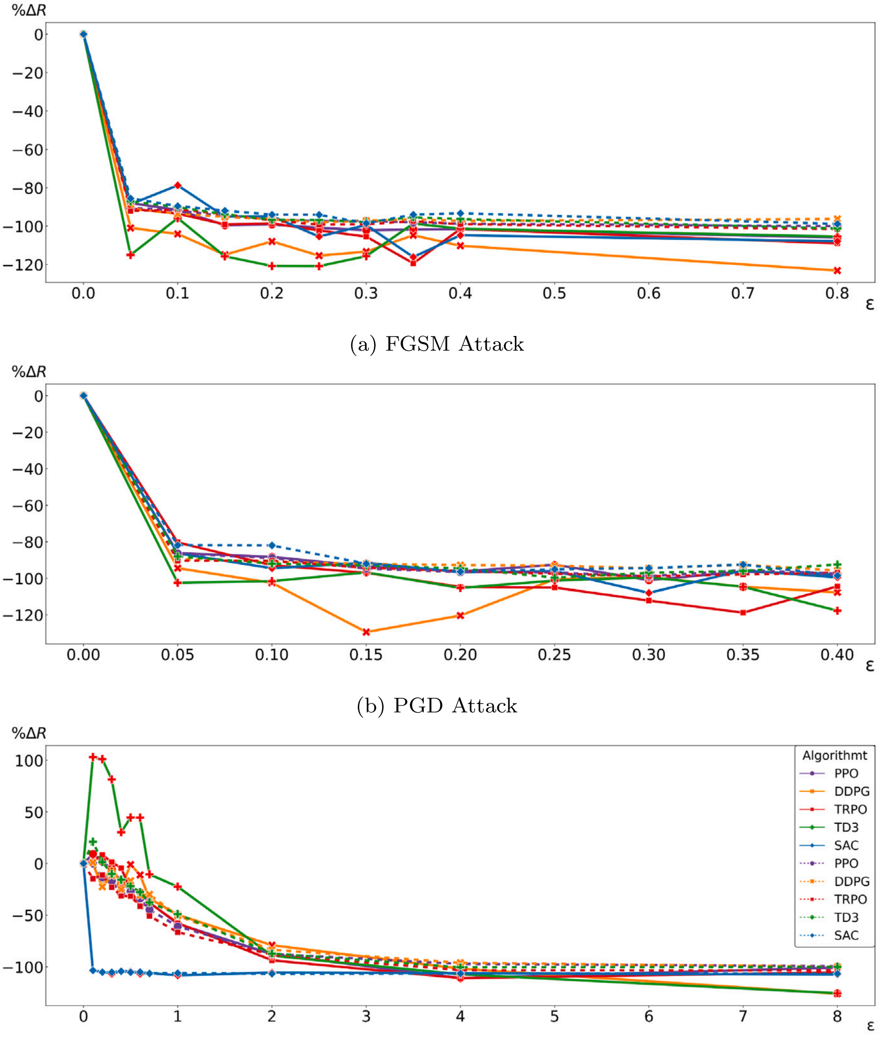


Fig. 5. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in HalfCheetah: This plot visualizes in detail the effect of increasing ϵ every 5% on $\Delta R\%$. Observe that the largest $\Delta R\%$ occurs for observation channel attacks when ϵ is low while the largest $\Delta R\%$ occurs for action channel attacks when ϵ is higher.

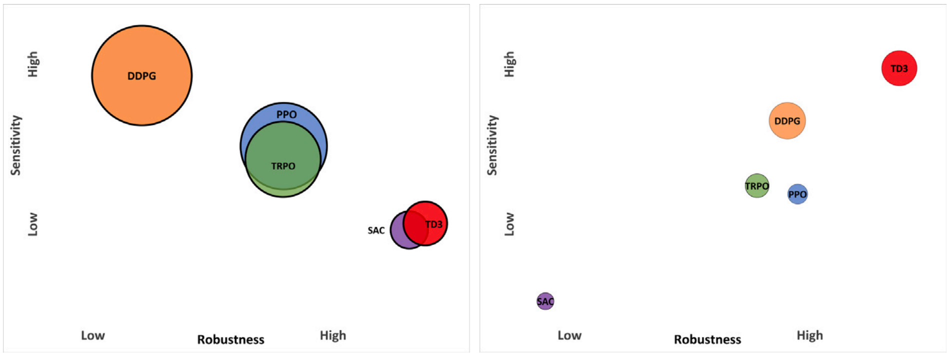


(a) FGSM Attack

(b) PGD Attack

(c) MAS Attack

Fig. 6. White-box attacks trends for HalfCheetah: The plots show the relationship between the value of ϵ (x-axis) and $\Delta R\%$ (y-axis). Line markers in the plots represent experiments we ran with a specific value of ϵ. The solid line represents the average $\Delta R\%$ across all environments, and the dotted line represents the average excluding the Ant environment.



(a) BlackBox Attack

(b) WhiteBox Attack

Fig. 7. Summary of observations: Visualization of the relative sensitivity and robustness of common RL benchmark algorithms. The x-axis denotes the robustness of the algorithms, and the y-axis denotes the sensitivity of the algorithms. Colors represent the five examined algorithms, the subplot on the left indicates the performance of the algorithm under black-box attacks, while the subplot on the right indicates the performance under white-box attacks. The circle sizes indicate the range of robustness by computing the differences between the maximum and the minimum $\Delta R\%$.

provide some initial insights. Fig. 6 illustrates the average $\Delta R\%$ cross all environments for each attack strategy. Similar to the black-box attacks we implemented, we observed that all white-box attacks resulted in a general negative trend. Compared to black-box attacks, we observed that the decrease in $\Delta R\%$ is much steeper than the trends observed in Fig. 4. However, it is worth highlighting that the context and range of the ϵ values used in white-box experiments are different. While the values of ϵ in the black-box experiments were expressed as a percentage of the action/observation space or the actual values of the action/observations themselves, the values of ϵ used in the white-box experiments were based on the values reported in the literature. Furthermore, the black-box attack strategies we proposed followed the strategy of adding noise, while the white-box strategy we implemented all incorporated some form of optimization. As such, no direct comparison between white and black-box trends can be made. Nevertheless, an interesting observation we made is that while increasing the value of ϵ resulted in a monotonic decrease in $\Delta R\%$ for black-box attacks, the $\Delta R\%$ for white-box attacks exhibited some form of fluctuations, although we still observe a general decreasing trend.

Comparing the attacks on the observation channel (FGSM and PGD) versus attacks on the action channel (MAS), we observe that, in general, both types of attacks perform similarly asymptotically as we increase the value of ϵ. However, at lower values of ϵ, action channel attacks have a higher variance in terms of the $\Delta R\%$ across different algorithms. Specifically, we observe that the smaller values of ϵ increased the $\Delta R\%$ by almost 100% for TD3. However, removing the results of the Ant environment from the trend (dotted lines) showed that the trend for TD3 reverts to a trend that follows the rest of the environment.

When we compare the performance across different RL policies, we observe that most of the algorithms had similar robustness, with DDPG and TD3 being the most sensitive to perturbations and displaying the largest $\Delta R\%$ when subjected to observation channel attacks. Once again, these trends became less extreme once removing the effect of the Ant environment. In terms of action channel attack (MAS), one interesting observation is that most RL algorithms performed similarly except for SAC, which displayed a large drop in performance even with a small value of ϵ. This is a surprising observation as SAC was one of the most robust policies in the black-box attack experiments, and even removing the effect of the Ant environment did not change the trends significantly. As such, it would be interesting for future studies to investigate why SAC is robust towards observation channel perturbation but becomes particularly sensitive to action space perturbation, specifically to the white-box MAS attack.

4.5. *Summary and discussions*

To summarize our findings, we compile our observations into Fig. 7 and rank the benchmark RL algorithms according to three criteria: robustness, range of robustness, and sensitivity. Furthermore, we classified the algorithm's characteristics according to black-box attacks in Fig. 7(a) and white-box attacks in Fig. 7(b). Formally, we define the sensitivity and robustness for black-box attacks as:

$$\begin{aligned}\text{Sensitivity} = \text{Average}(&|\Delta R_{10}\% - \Delta R_{5}\%| + |\Delta R_{15}\% - \Delta R_{10}\%| \\ &+ |\Delta R_{20}\% - \Delta R_{15}\%| + |\Delta R_{25}\% - \Delta R_{20}\%| \\ &+ |\Delta R_{50}\% - \Delta R_{25}\%|)\end{aligned} \tag{2}$$

$$\text{Robustness} = \text{Average}(\Delta R_{5}\% + \Delta R_{10}\% + \Delta R_{15}\% + \Delta R_{20}\% + \Delta R_{25}\% + \Delta R_{50}\%) \tag{3}$$

where the notation $\Delta R_5\%$ represents the value of percentage change in rewards ($\Delta R\%$) due to a black-box attack with $\epsilon = 5\%$. Similarly, we define the sensitivity and robustness for white-box attacks as:

$$\begin{aligned}\text{Sensitivity} = \text{Average}(&|\Delta R_{MAS}\% - \Delta R_{FGSM}\%| \\ &+ \ |\Delta R_{MAS}\% - \Delta R_{PGD}\%| + |\Delta R_{FGSM}\% - \Delta R_{PGD}\%|)\end{aligned} \tag{4}$$

$$\text{Robustness} = \text{Average}(\Delta R_{MAS}\% + \Delta R_{FGSM}\% + \Delta R_{PGD}\%) \tag{5}$$

In Figs. 7(a) and 7(b), the horizontal axis represents an algorithm's robustness, where we define robustness as the average $\Delta R\%$ across all attacks and all environments, as shown in Eqs. (3) and (5). The vertical axis represents an algorithm's sensitivity. We define sensitivity by taking the average difference for all $\Delta R\%$ across all possible pairs of strategies and computing its absolute value, as shown in Eqs. (2) and (4). Intuitively, the sensitivity gives us a sense of how much we can expect the performance of an RL policy will change when subjected to different attacks. Finally, the size of the circles in Fig. 7 represents the range of the robustness of an algorithm. We define the range of robustness by taking the difference between the maximum and minimum $\Delta R\%$ under the white-box and black-box attack scenarios, respectively:

$$\text{Range of Robustness} = \max(\Delta R_{\mathcal{A}}\%) - \min(\Delta R_{\mathcal{A}}\%) \tag{6}$$

where $\mathcal{A}$ represents the set of all attacks under the black-box and white-box scenarios respectively. Generally speaking, we observe that TD3 exhibits the best robustness across both white-box and black-box attacks, while SAC performs well under black-box attacks but performs

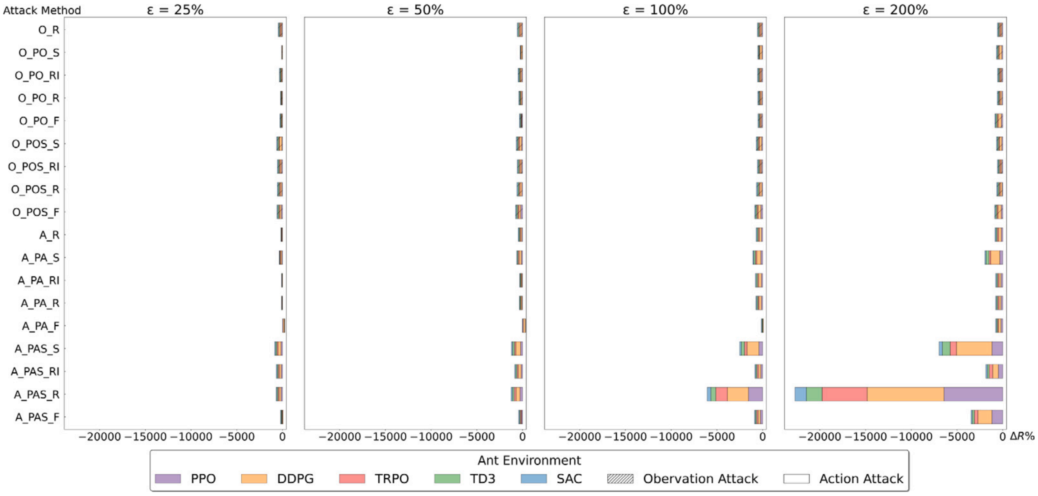


Fig. A.8. Comparison of black-box attack on Ant:: All black-box strategies are shown on the *y*-axis, and the *x*-axis represents the cumulative $\Delta R\%$ across all RL algorithms. The algorithms are present by the colors. The shaded bar and solid bar show the observation and the action channel. Each subplot represents a particular attack budget ϵ.

extremely poorly on white-box attacks. We also note that PPO and TRPO are robust to a certain extent with medium sensitivities, but DDPG ranks the lowest in terms of having low robustness and high sensitivity. Finally, we also highlight that black-box attacks have a larger range of effects on the RL policies in general (larger circles) when compared to white-box attacks, which have more consistent effects (smaller circles).

5. Conclusion

In this work, we compared commonly used benchmark RL algorithms' robustness towards various types of perturbation during test time. We designed a suite of simple black-box attack strategies to perturb the RL agent's observation and action channels, and we also implemented three commonly used white-box optimization-based attacks that perturbed the agent's observation and action channels. From our experiments, we made the following conclusions: Firstly, from the black-box attack strategies we tested, a recurring theme is that observation channel attacks are more effective than action channel attacks, but only until a certain threshold on the magnitude of the perturbation. Beyond this threshold, the effects of observation attacks saturate while action channel attacks may continue to have some effect. We also find that the Ant environment generally amplifies the effect of attacks. In terms of the robustness of different policies under black-box attacks, SAC and TD3 were generally robust, while DDPG and TRPO were the most sensitive to perturbations. When subjected to optimization-based white-box attacks in the observation channel, most policies performed similarly, with DDPG and TD3 being the most sensitive, while SAC was found to be extremely sensitive to action channel attacks. We find it intriguing that two of the most robust policies under black-box attacks ended up being the most sensitive to attacks under white-box attacks, and future work will seek to further understand this phenomenon. Furthermore, we will extend this study to include a more comprehensive comparison of existing optimization-based black-box attacks and white-box attacks.

CRediT authorship contribution statement

Qisai Liu: Conceptualization, Data curation, Formal analysis, Methodology, Validation, Writing – original draft, Writing – review & editing. **Xian Yeow Lee:** Validation, Writing – review & editing. **Soumik Sarkar:** Project administration, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Codes to reproduce the results can be found at: https://github.com/super864/Natural-Robustness-RL.

Acknowledgments

QSL and XYL contributed to conducting the idea and concluding the results, and QSL also contributed to implementing the experiment. SS contributed to evaluating the paper and analyzing the results. All of the authors contributed to writing the paper and reading it.

Funding

This work was partly supported by the National Science Foundation, United States under grants CAREER-1845969.

Appendix

In this supplementary appendix section, we present a comprehensive collection of comparison plots for the remaining environments that were not showcased within the main manuscript. The purpose of including these additional plots is to offer a more in-depth analysis and a broader understanding of the experimental outcomes. By examining the diverse range of environments, we aim to provide a comprehensive overview of the results, ensuring a comprehensive evaluation of the proposed methodology. The inclusion of these supplementary plots serves to enhance the scientific rigor of the research and provides readers with a more complete picture of the experimental findings (see Fig. A.25).

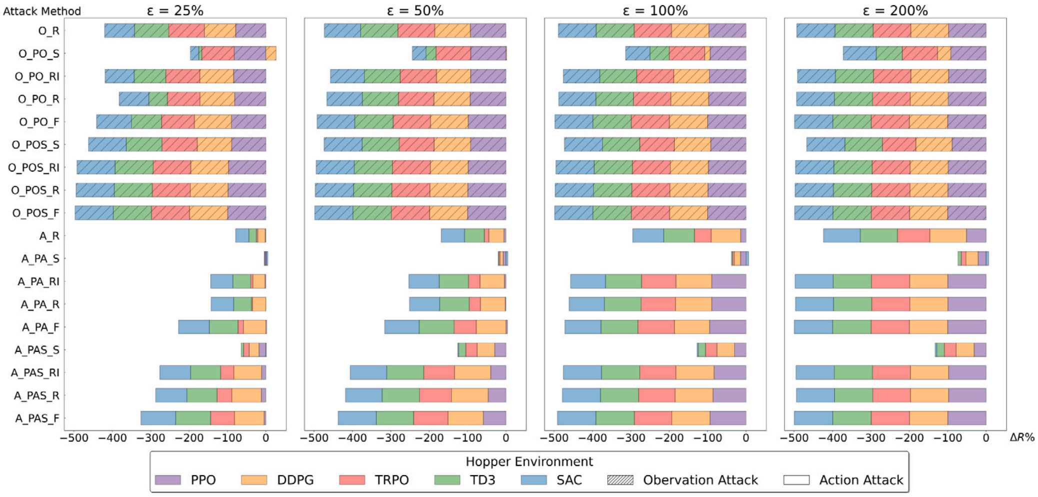


Fig. A.9. Comparison of black-box attack on Hopper.

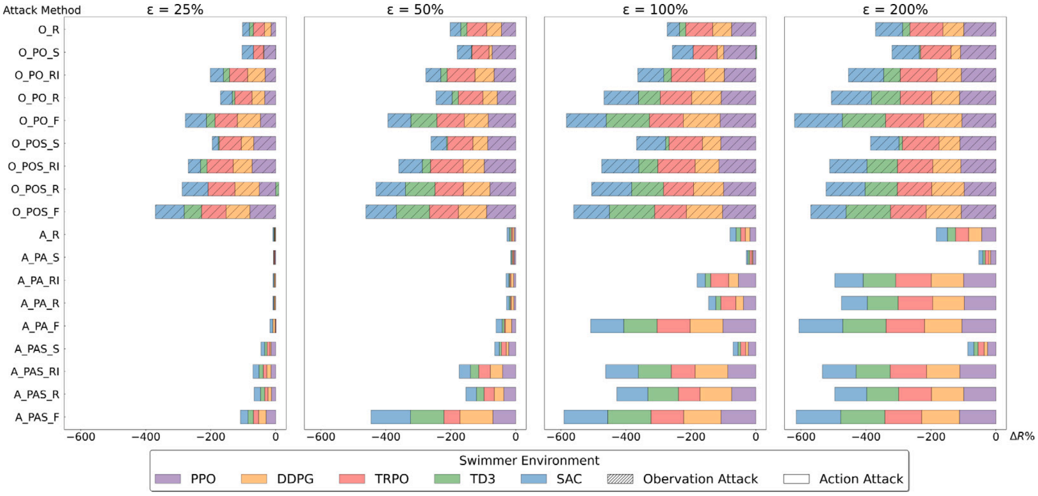


Fig. A.10. Comparison of black-box attack on Swimmer.

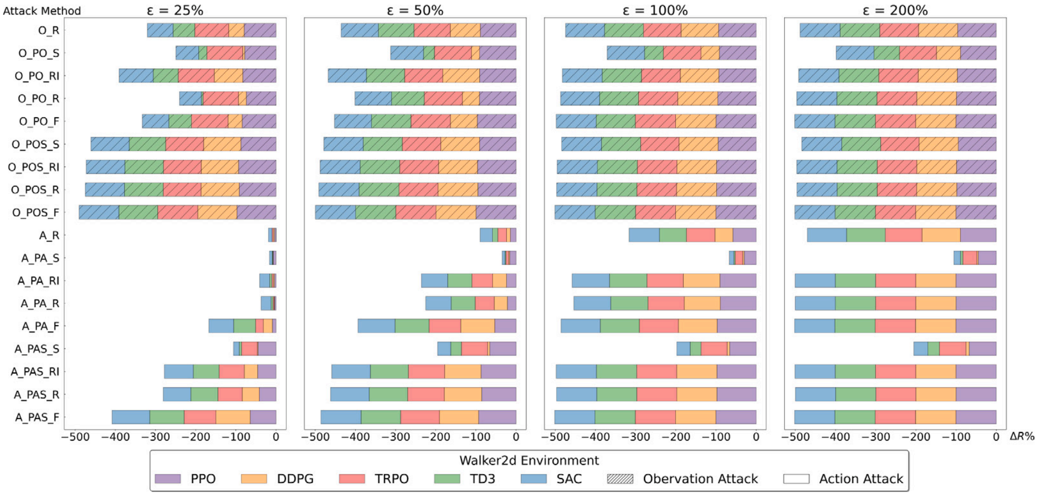


Fig. A.11. Comparison of black-box attack on Walker.

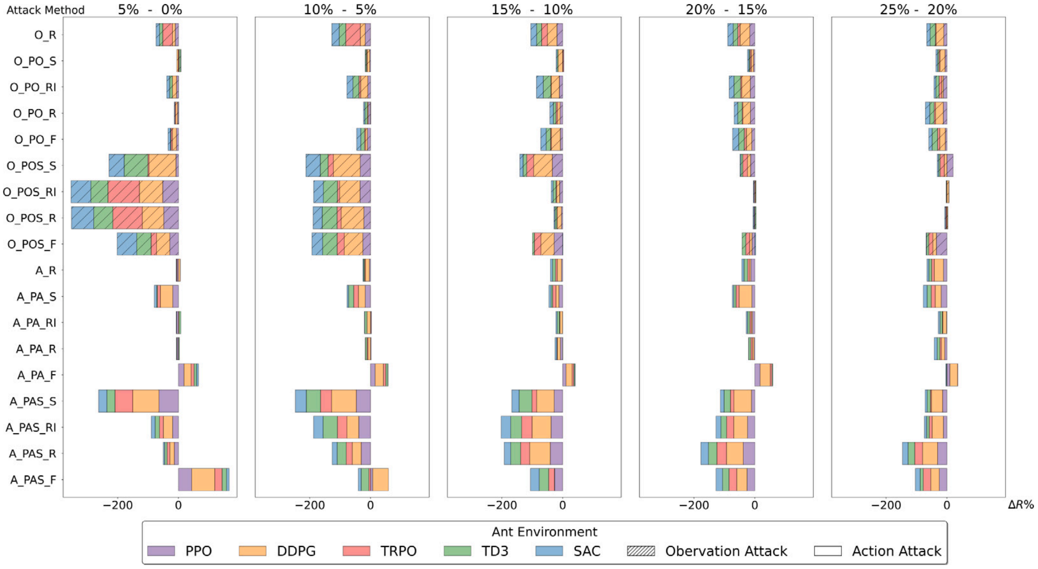


Fig. A.12. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Ant.

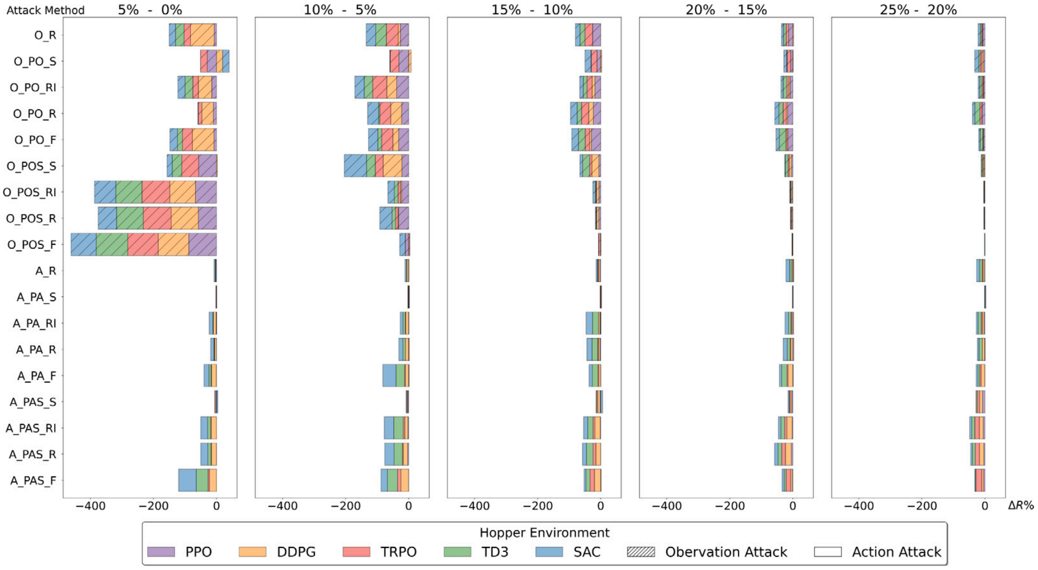


Fig. A.13. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Hopper.

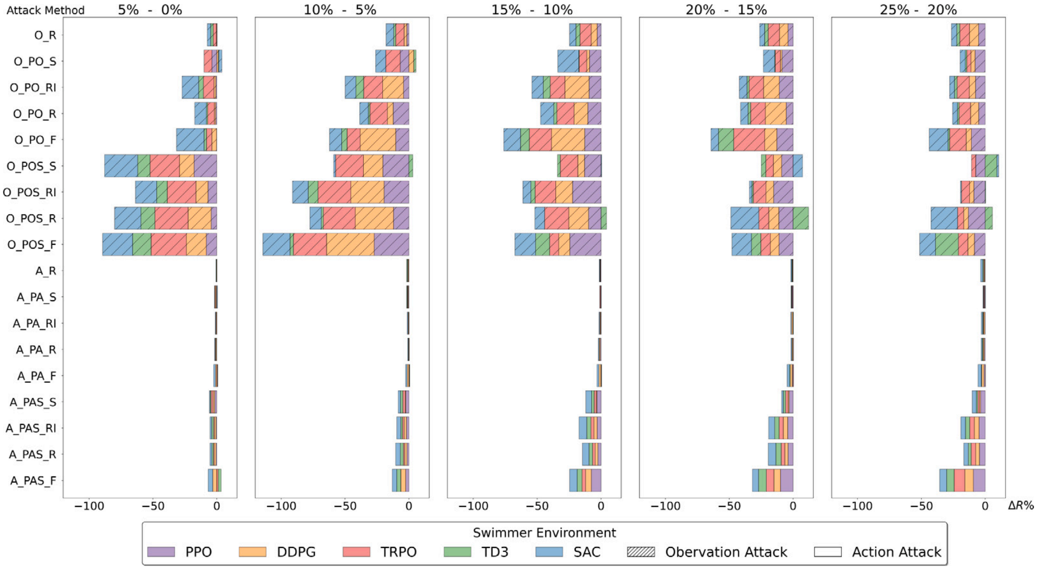


Fig. A.14. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Swimmer.

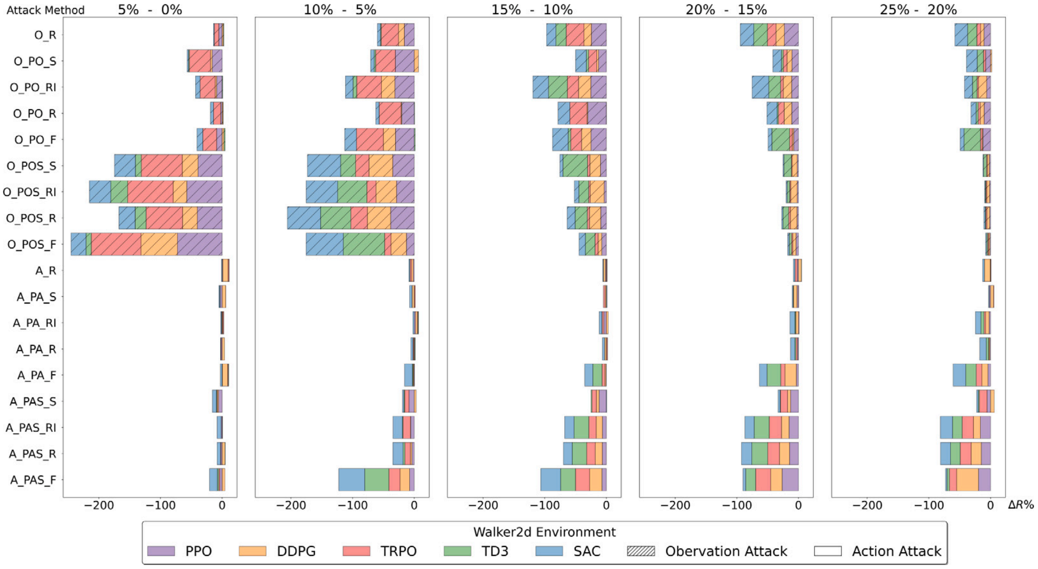


Fig. A.15. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Walker.

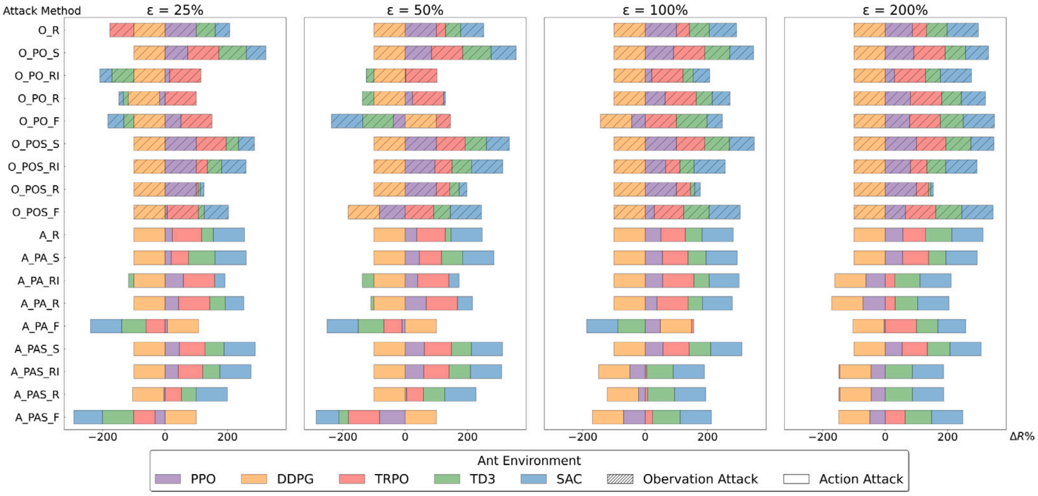


Fig. A.16. Comparison of black-box attacks on Ant with normalization: This figure illustrates the performance of various black-box attack strategies against the Ant environment, with each strategy normalized between −1 to 1 for five unique policies. The y-axis details the attack strategies, whereas the x-axis shows the cumulative percentage change in rewards $\Delta R\%$ across diverse reinforcement learning algorithms, each distinguished by color. Shaded and solid bars represent the observation and action channels, respectively. Each subplot highlights a specific attack budget ϵ, thoroughly examining attack impacts across different scenarios.

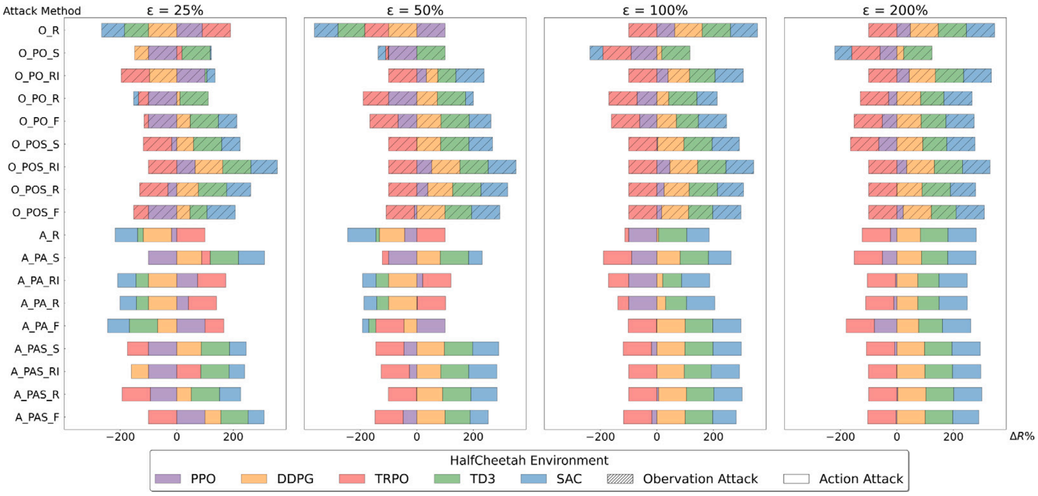


Fig. A.17. Comparison of black-box attacks on HalfCheetah with normalization.

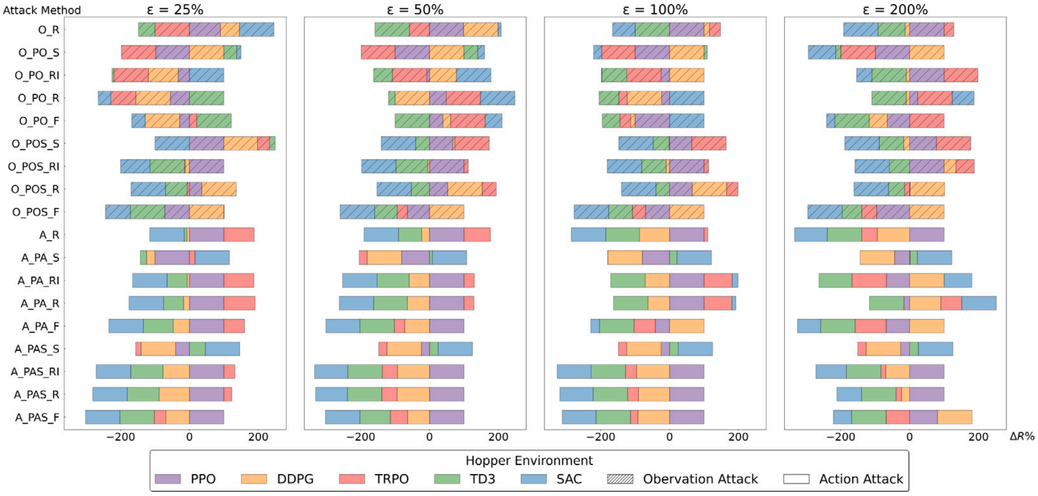


Fig. A.18. Comparison of black-box attacks on Hopper with normalization.

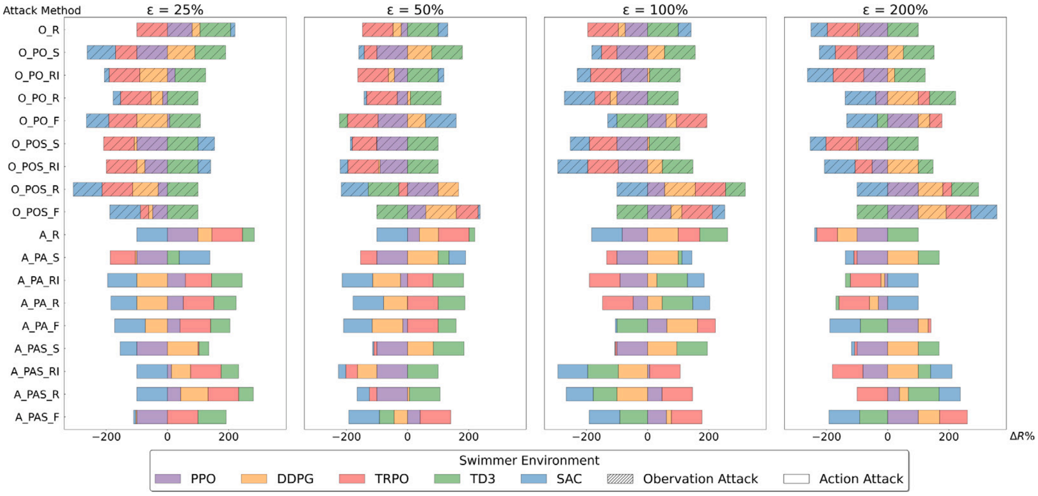


Fig. A.19. Comparison of black-box attacks on Swimmer with normalization.

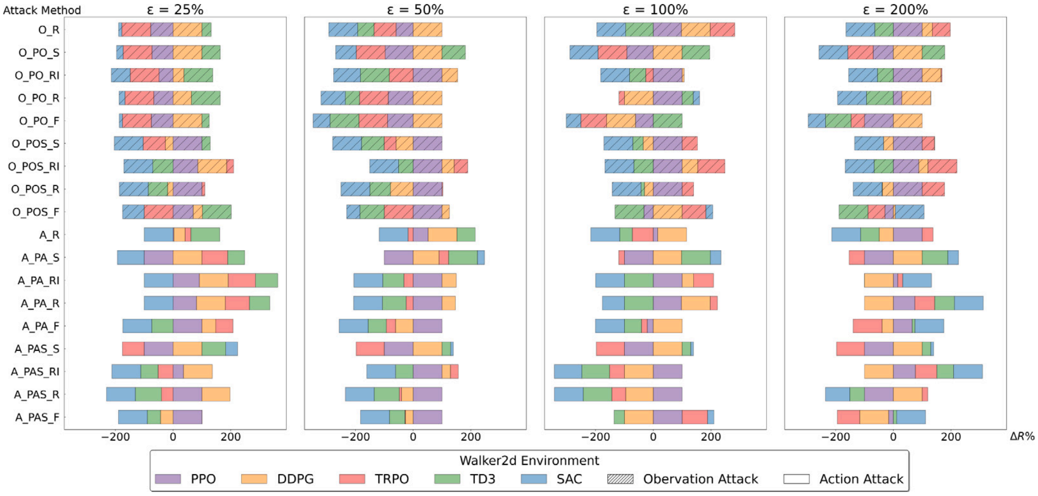


Fig. A.20. Comparison of Black-Box Attacks on Walker with normalization.

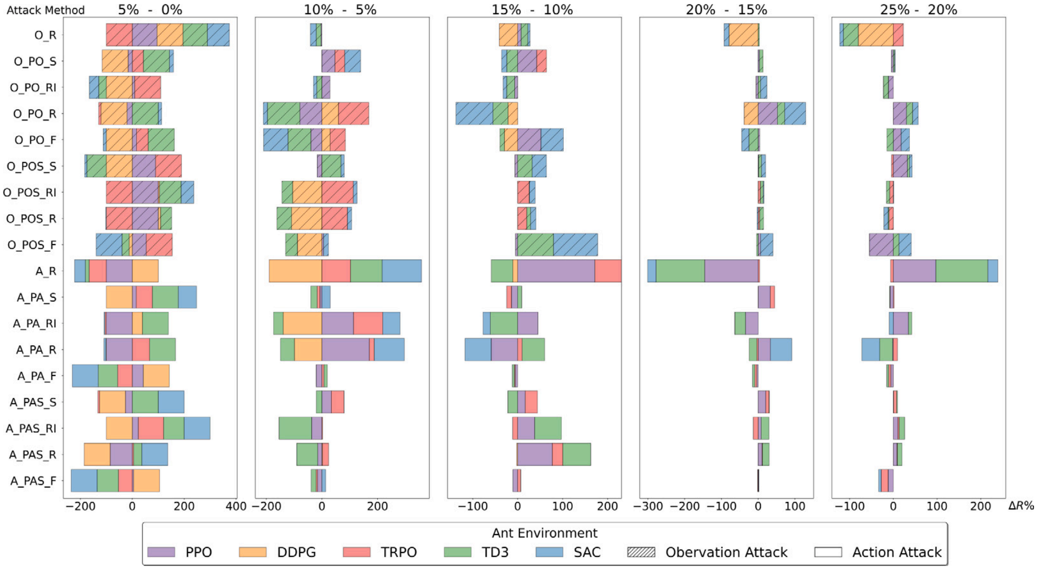


Fig. A.21. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Ant with normalization: This figure precisely showcases the influence of incremental ϵ adjustments, every 5%, on the $\Delta R\%$, considering various policies. It normalizes individual attack strategies within a −1 to 1 range, clearly depicting each policy's impact.

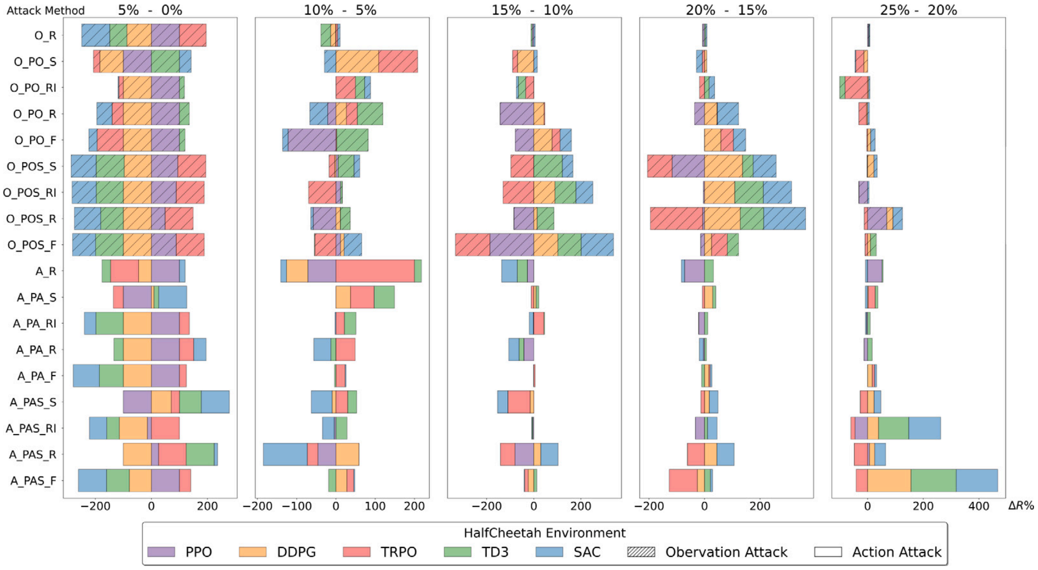


Fig. A.22. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in HalfCheetah with normalization.

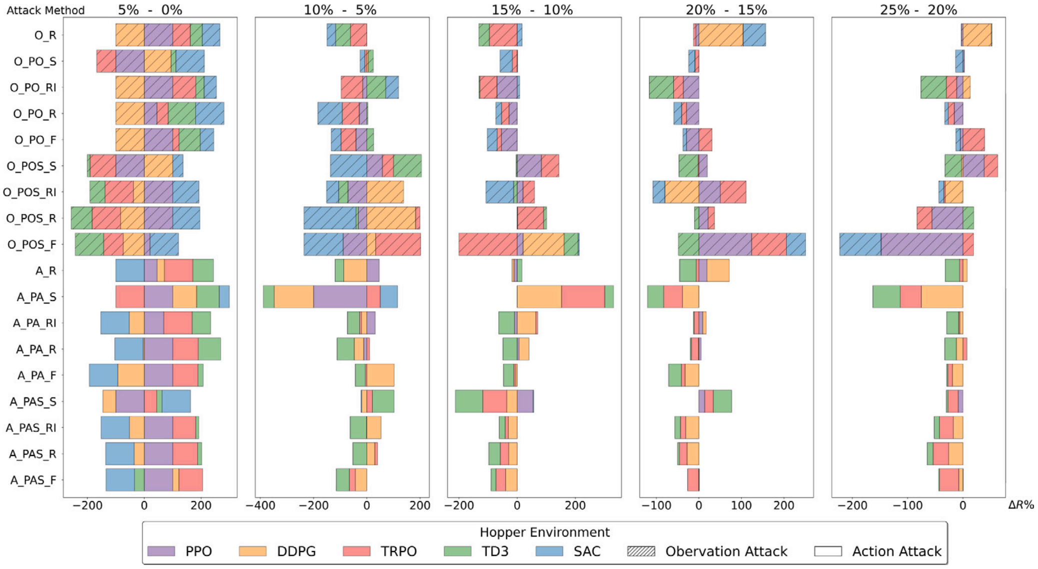


Fig. A.23. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Hopper with normalization.

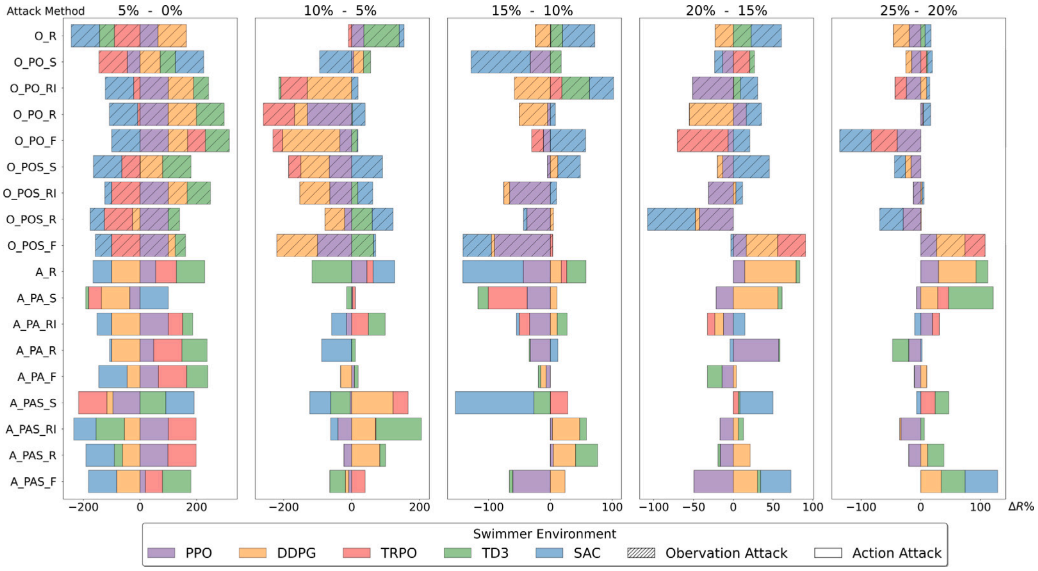


Fig. A.24. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Swimmer with normalization.

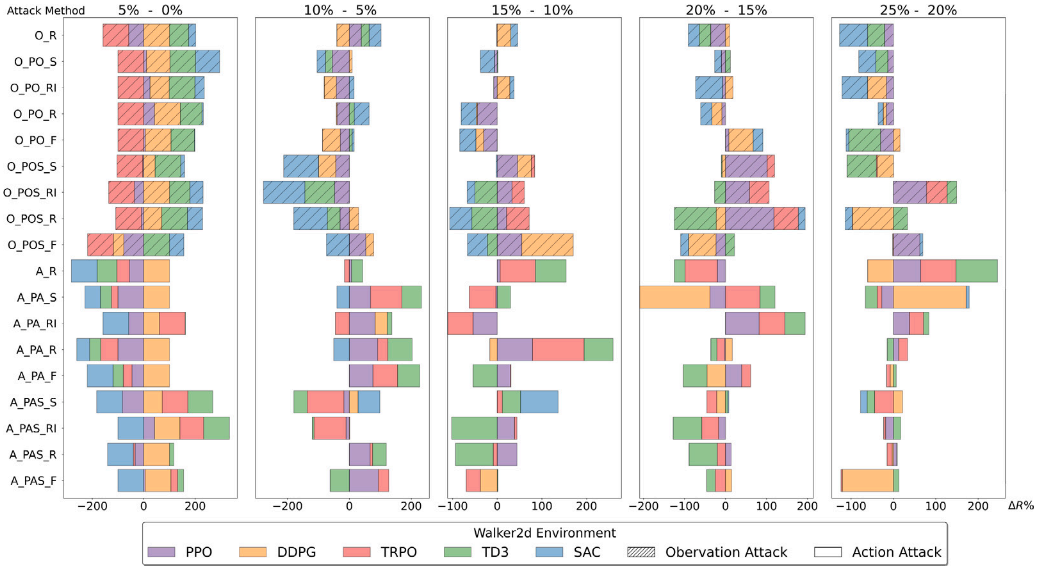


Fig. A.25. Detailed visualization of the effect of increasing ϵ on $\Delta R\%$ in Walker with normalization.

References

Abbeel, P., Coates, A., Quigley, M., Ng, A., 2006. An application of reinforcement learning to aerobatic helicopter flight. In: Schölkopf, B., Platt, J., Hoffman, T. (Eds.), Advances in Neural Information Processing Systems, vol. 19. MIT Press, URL https://proceedings.neurips.cc/paper/2006/file/98c39996bf1543e974747a2549b3107c-Paper.pdf.

Argawal, R., Kalel, D., Harshit, M., Domnic, A.D., Singh, R.R., 2021. Sensor fault detection using machine learning technique for automobile drive applications. In: 2021 National Power Electronics Conference. NPEC, IEEE, pp. 1–6.

Bahrin, M.A.K., Othman, M.F., Azli, N.H.N., Talib, M.F., 2016. Industry 4.0: A review on industrial automation and robotic. J. Teknol. 78 (6–13).

Behzadan, V., Munir, A., 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In: International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, pp. 262–275.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W., 2016. OpenAI Gym. arXiv preprint arXiv:1606.01540.

Carlini, N., Liu, C., Kos, J., Erlingsson, Ú., Song, D., 2019. The secret sharer: Measuring unintended neural network memorization & extracting secrets. arXiv preprint arXiv:1802.08232.

Chen, T., Liu, J., Xiang, Y., Niu, W., Tong, E., Han, Z., 2019. Adversarial attack and defense in reinforcement learning-from AI security view. Cybersecurity 2 (1), 1–22.

Ebrahimi, J., Rao, A., Lowd, D., Dou, D., 2017. Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751.

Fujimoto, S., Hoof, H., Meger, D., 2018. Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning. PMLR, pp. 1587–1596.

Fujita, Y., Nagarajan, P., Kataoka, T., Ishikawa, T., 2021. ChainerRL: A deep reinforcement learning library. J. Mach. Learn. Res. 22 (77), 1–14, URL http://jmlr.org/papers/v22/20-376.html.

Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., Russell, S., 2020. Adversarial policies: Attacking deep reinforcement learning. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=HJgEMpVFwB.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Guo, C., Gardner, J., You, Y., Wilson, A.G., Weinberger, K., 2019. Simple black-box adversarial attacks. In: International Conference on Machine Learning. PMLR, pp. 2484–2493.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al., 2018. Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., Abbeel, P., 2017. Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284.

Jan, S.U., Lee, Y.D., Koo, I.S., 2021. A distributed sensor-fault detection and diagnosis framework using machine learning. Inform. Sci. 547, 777–796.

Lee, X.Y., Ghadai, S., Tan, K.L., Hegde, C., Sarkar, S., 2020. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4577–4584.

Li, X., Li, Y., Feng, Z., Wang, Z., Pan, Q., 2023. ATS-O2A: A state-based adversarial attack strategy on deep reinforcement learning. Comput. Secur. 129, 103259.

Li, Y., Pan, Q., Cambria, E., 2022. Deep-attack over the deep reinforcement learning. Knowl.-Based Syst. 250, 108965.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., Sun, M., 2017. Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748.

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

Mahmood, K., Mahmood, R., Rathbun, E., van Dijk, M., 2021. Back in black: A comparative evaluation of recent state-of-the-art black-box attacks. IEEE Access 10, 998–1019.

Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., Peters, J., 2022. Robust reinforcement learning: A review of foundations and recent advances. Mach. Learn. Knowl. Extr. 4 (1), 276–315.

Papernot, N., McDaniel, P., Goodfellow, I., 2016a. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277.

Papernot, N., McDaniel, P., Jha, S., 2017. The limitations of deep learning in adversarial settings. In: 2017 IEEE European Symposium on Security and Privacy. EuroS&P, IEEE, pp. 372–387.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A., 2016b. The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy. EuroS&P, IEEE, pp. 372–387.

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

Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P., 2015. Trust region policy optimization. In: International Conference on Machine Learning. PMLR, pp. 1889–1897.

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

Sun, Y., Zheng, R., Liang, Y., Huang, F., 2022. Who is the strongest enemy? Towards optimal and efficient evasion attacks in deep RL. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=JM2kFbJvvI.

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

Tan, K.L., Esfandiari, Y., Lee, X.Y., Sarkar, S., et al., 2020. Robustifying reinforcement learning agents via action space adversarial training. In: 2020 American Control Conference. ACC, IEEE, pp. 3959–3964.

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

Wang, Y., Liu, H., Zheng, W., Xia, Y., Li, Y., Chen, P., Guo, K., Xie, H., 2019. Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning. IEEE Access 7, 39974–39982. http://dx.doi.org/10.1109/ACCESS.2019.2902846.

Xiao, C., Pan, X., He, W., Peng, J., Sun, M., Yi, J., Liu, M., Li, B., Song, D., 2019. Characterizing attacks on deep reinforcement learning. arXiv preprint arXiv:1907.09470.

Yang, J., Jiang, Y., Huang, X., Ni, B., Zhao, C., 2020. Learning black-box attackers with transferable priors and query feedback. Adv. Neural Inf. Process. Syst. 33, 12288–12299.

Zhang, H., Chen, H., Boning, D., Hsieh, C.-J., 2021. Robust reinforcement learning on state observations with learned optimal adversary. arXiv preprint arXiv:2101.08452.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., Hsieh, C.-J., 2020. Robust deep reinforcement learning against adversarial perturbations on state observations. Adv. Neural Inf. Process. Syst. 33, 21024–21037.