Improve Robustness of Reinforcement Learning against Observation Perturbations via l_∞ Lipschitz Policy Networks

Buqing Nie, Jingtian Ji, Yangqing Fu, Yue Gao*

MoE Key Lab of Artificial Intelligence and AI Institute, Shanghai Jiao Tong University
{niebuqing, jijingtian, frank79110, yuegao}@sjtu.edu.cn

Abstract

Deep Reinforcement Learning (DRL) has achieved remarkable advances in sequential decision tasks. However, recent works have revealed that DRL agents are susceptible to slight perturbations in observations. This vulnerability raises concerns regarding the effectiveness and robustness of deploying such agents in real-world applications. In this work, we propose a novel robust reinforcement learning method called *SortRL*, which improves the robustness of DRL policies against observation perturbations from the perspective of the network architecture. We employ a novel architecture for the policy network that incorporates global l_∞ Lipschitz continuity and provide a convenient method to enhance policy robustness based on the output margin. Besides, a training framework is designed for *SortRL*, which solves given tasks while maintaining robustness against l_∞ bounded perturbations on the observations. Several experiments are conducted to evaluate the effectiveness of our method, including classic control tasks and video games. The results demonstrate that *SortRL* achieves state-of-the-art robustness performance against different perturbation strength.

Introduction

Recently, Deep Reinforcement Learning (DRL) has achieved breakthrough success in various application scenarios, including video games (Mnih et al. 2015), recommender systems (Afsar, Crump, and Far 2022), and robotics control (Lee et al. 2020). These achievements typically rely on the Deep Neural Networks (DNNs) as function approximators for their strong expressive power, which enables the end-to-end learning of policies in complex environments with high-dimension state spaces, such as images observations (Hornik, Stinchcombe, and White 1989; Mnih et al. 2015; Kaiser et al. 2020).

However, DNNs typically lack robustness due to their highly non-linear and black-box nature, resulting in unreasonable and unpredictable outputs when inputs are perturbed slightly (Madry et al. 2018; Yuan et al. 2019). Similarly, recent works have shown that typical DNN-based policies are also vulnerable to imperceptible perturbations on observations, also known as "state adversaries", which are prevalent in application scenarios such as sensor noise (Zang et al. 2019) and adversarial attacks (Huang et al. 2017). These slight perturbations can deceive typical DRL policies easily, leading to irrational and unpredictable decisions by the agent (Fischer et al. 2019; Zhang et al. 2020b; Oikarinen et al. 2021; Zhang et al. 2021b; Sun et al. 2022). This may affect the policy effectiveness and user experience, even causing safety issues, especially in safety-critical applications such as autonomous driving and robot manipulation tasks (Zhao et al. 2022). The lack of robustness to observation perturbations renders applications of DRL unreliable and risky, thereby limiting potential applications in real-world scenarios.

In the recent decade, plenty of works have been proposed to certify and enhance the robustness of DRL policies against perturbations on observations. Some researchers propose various robust policy regularizers to enforce policy smoothness, i.e. the policy output similar actions given similar observations (Zhang et al. 2020b; Shen et al. 2020; Oikarinen et al. 2021). For example, Shen et al. (Shen et al. 2020) propose a smoothness-inducing regularizer inspired by Lipschitz continuity to encourage the policy function to become smooth, which improves sample efficiency and policy robustness in continuous control tasks. Despite the excellent performance achieved, the incorporation of a smoothness regularizer may hinder the expressive power of the policy network, resulting in a partial compromise of optimality and performance, especially in tasks with strong perturbation strength (Wu and Vorobeychik 2022).

Another approach to enhancing the policy robustness is based on attacking and adversarial samples (Mandlekar et al. 2017; Pattanaik et al. 2018; Zhang et al. 2021b). For instance, Pattanaik et al. (Pattanaik et al. 2018) improve policy robustness utilizing adversarial observations found by gradient-based attackers. Recently, Zhang et al. (Zhang et al. 2021b) propose Alternating Training with Learned Adversaries (ATLA), which trains an RL adversary online with the agent policy alternately. ALTA significantly improves the policy robustness in continuous control tasks. Despite the excellent robustness, these methods require training extra attackers or finding adversaries for the observations, which incurs additional computational and sampling costs, thereby limiting their practical applications.

In this work, we propose a novel method called *SortRL* to

*Corresponding author.

improve the robustness of DRL policies against observation perturbations from the perspective of the network architecture. We introduce a new policy network architecture based on an l_∞ Lipschitz Neural Network called *SortNet*. Besides, we introduce a straightforward and efficient method to estimate the lower bound of policy robustness utilizing the output margin. Additionally, we design a training framework for *SortRL* based on Policy Distillation (Rusu et al. 2016), which enables the agent to solve the given tasks successfully while addressing robustness requirements against observation perturbations. Several experiments on classic control tasks and video games are conducted to evaluate the performance of *SortRL*, which demonstrates the state-of-the-art performance of our method.

Our main contributions are listed as follows:

- We propose a novel robust reinforcement learning method called *SortRL*, which enhances the policy robustness against observation perturbations. To our knowledge, this is the first work to address this issue from the perspective of network architecture.
- We employ a novel policy design base on an l_∞ Lipschitz Neural Network. A convenient method is provided to evaluate and improve policy robustness based on the output margin.
- We design a training framework for *SortRL* to make a trade-off between optimality and robustness, which enables the agent to solve given tasks while addressing robustness requirements.
- Experiments on classic control tasks and video games are conducted, which demonstrate that *SortRL* achieves state-of-the-art robustness against different perturbation strength, especially in tasks with strong perturbations.

Related Work

Robust Reinforcement Learning

Robust Reinforcement Learning aims to improve the policy robustness against perturbations in the Markov Decision Process (MDP). Thus, there exist various interpretations of robustness in the RL context, including the robustness against action perturbations (Tessler, Efroni, and Mannor 2019), dynamics uncertainty (Pinto et al. 2017; Huang et al. 2022), domain shift (Muratore, Gienger, and Peters 2019; Ju et al. 2022), and reward perturbations (Wang, Liu, and Li 2020; Eysenbach and Levine 2021).

This work focuses on the policy robustness against observation perturbations, which has been actively researched recently (Fischer et al. 2019; Zhang et al. 2020b; Oikarinen et al. 2021; Liang et al. 2022). Several works improve robustness against observation perturbations utilizing various policy regularizers, which enforce the policy to make similar decisions under similar observations (Zhang et al. 2020b; Shen et al. 2020; Oikarinen et al. 2021). For instance, Shen et al. (Shen et al. 2020) design a policy regularizer for continuous control tasks inspired by the Lipschitz continuity, which improves sample efficiency and robustness to adversarial perturbations. Some researchers attempt to enforce policy robustness utilizing adversarial samples generated through active attacks (Mandlekar et al. 2017; Pattanaik et al. 2018; Zhang et al. 2021b; Liang et al. 2022). Zhang et al. (Zhang et al. 2021b) propose ATLA, which improves the policy robustness in continuous control tasks by training the policy with an RL adversary online together. However, Korkmaz (Korkmaz 2021, 2023) points out that adversarially trained DRL policies may still be sensitive to policy-independent perturbations. Several researchers study the certified robustness of DRL policies (Fischer et al. 2019; Everett, Lütjens, and How 2021). Some methods such as CROP (Wu et al. 2022) and Policy Smoothing (Kumar, Levine, and Feizi 2022) are proposed to analyze robustness certificates for trained DRL policies.

Despite the significant achievements, there are still some limitations to be addressed. For instance, they may suffer from high computational costs (Zhang et al. 2021b) and struggle to cope with strong perturbations, such as perturbations strength greater than $5/255$ in video games (Wu and Vorobeychik 2022). In this work, we propose a new robust RL method called *SortRL*. To our knowledge, this is the first method to improve the robustness of RL policies against observation perturbations from the perspective of network architecture.

Robustness of Neural Networks

Standard neural networks are vulnerable to small perturbations to the inputs (Szegedy et al. 2014; Madry et al. 2018), especially given high dimensional inputs such as images. In order to improve the robustness of DNN, various methods are proposed, including randomized smoothing (Salman et al. 2019) and relaxation-based approaches (Gowal et al. 2018; Zhang et al. 2020a). Besides, some researchers have found that the Lipschitz continuity is significant to the network robustness (Tsuzuku, Sato, and Sugiyama 2018; Anil, Lucas, and Grosse 2019; Li et al. 2019). Recently, several Lipschitz Neural Networks (LNN) have been proposed to enhance robustness, including Spectral Norm (Gouk et al. 2021), GroupSort (Anil, Lucas, and Grosse 2019), and l_∞-distance neuron (Zhang et al. 2022a, 2021a). In this work, we construct the policy network based on an l_∞ 1-Lipschitz Neural Network called SortNet (Zhang et al. 2022b), which provides Lipschitz property, strong expressive power, and high computation efficiency.

Methodology

Problem Formulation

To study policy robustness under observation perturbations, we formulate the decision process based on the state-adversarial Markov Decision Process (SA-MDP) (Zhang et al. 2020b). In this work, an SA-MDP $\widetilde{\mathcal{M}}$ is defined as $< \mathcal{S}, \mathcal{A}, P, R, \gamma, \rho, \nu >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ denotes the action space, $P(s'|s,a) = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$ denotes the transition probability, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the reward function, $\gamma \in [0, 1]$ denotes discount factor, and $\rho(s) = \Pr(s_0)$ is the distribution of initial states. $\pi : \mathcal{S} \to \Pr(\mathcal{A})$ is a stationary policy, which is trained to maximize the cumulative reward.

Different from typical MDP $\mathcal{M}$, there exists an adversary $\nu(s) : \mathcal{S} \to \Pr(\mathcal{S})$ in SA-MDP $\widetilde{\mathcal{M}}$, which adds perturbations to the agent's observations. Each time the agent obtains perturbed observation $\hat{s} \sim \nu(s)$ and makes the decision $a \sim \pi(\cdot|\hat{s})$. Therefore, the value function of policy π under ν adversary is given as follows:

$$\widetilde{V}_{\pi\circ\nu}(s) = \mathbb{E}_{\hat{s}_t\sim\nu(s_t),a_t\sim\pi(\hat{s}_t)}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}|s_0 = s\right]. \quad (1)$$

In this work, we focus on RL tasks with discrete action spaces against l_∞ bounded perturbations, i.e. $\nu(s) \in \mathcal{B}_\epsilon^\infty(s)$, where $\mathcal{B}_\epsilon^\infty(s) = \{\hat{s}|\,\|\hat{s} - s\|_\infty \le \epsilon\}$ denotes the "neighbors" of the clean state s. The $\epsilon \ge 0$ is an important parameter determining the strength of the adversary. A larger value of ϵ indicates a stronger adversary, which in turn requires a higher level of policy robustness. Thus, the policy π can be trained by solving the following optimization problem:

$$\begin{aligned}&\max_{\pi}\min_{\nu}\mathbb{E}_{s\sim\rho}\left[\widetilde{V}_{\pi\circ\nu}(s)\right]\\ &\text{s.t.}\quad \|\hat{s} - s\|_\infty \le \epsilon, \forall s \in \mathcal{S},\ \hat{s} \sim \nu(s).\end{aligned} \quad (2)$$

Problem Transformation

We are required to solve a minimax optimization problem as described in Eq. (2). However, finding the optimal adversary $\nu^*(s) = \arg\min_\nu \widetilde{V}_{\pi\circ\nu}(s)$ for each state s_t is NP-hard, which is computationally and sample expensive (Oikarinen et al. 2021). To address this issue, we try to reformulate the problem in this section.

Theorem 1. *Given a typical MDP $\mathcal{M}$, corresponding SA-MDP $\widetilde{\mathcal{M}}$ with an adversary $\nu(s) \in \mathcal{B}_\epsilon^\infty(s)$, and a policy π, $V_\pi(s)$ and $\widetilde{V}_{\pi\circ\nu}(s)$ denote the value functions in $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ accordingly. We have:*

$$\begin{aligned}&\max_{s\in\mathcal{S}}\{V_\pi(s) - \min_\nu \widetilde{V}_{\pi\circ\nu}(s)\}\\ &\le \alpha\max_{s\in\mathcal{S}}\max_\nu\sqrt{D_{\mathrm{KL}}(\pi(s),\pi(\hat{s}))},\end{aligned} \quad (3)$$

where $\alpha = \sqrt{2}\left[1 + \frac{\gamma}{(1-\gamma)^2}\right]\max_{(s,a,s')}|R(s,a,s')|$ is a constant independent of the policy, $\hat{s} \sim \nu(s)$ denotes perturbed observation, and $D_{\mathrm{KL}}(\cdot,\cdot)$ denotes KL-divergence.

The proof is given in Appendix A.1 according to (Achiam et al. 2017) and (Zhang et al. 2020b). Theorem 1 indicates that the performance loss of the policy π under the optimal adversary ν^* is bounded by the KL divergence between the action distributions. Therefore, in order to minimize the performance loss of π against the observation adversary, we can minimize the D_{KL} illustrated in Eq. (3) during training. One possible approach is constructing policy regularizers based on D_{KL}, such as $\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_s\left[\max_\nu D_{\mathrm{KL}}\left(\pi\left(s\right),\pi\left(\hat{s}\right)\right)\right]$, which is minimized during training the policy. However, finding the adversary $\arg\max_\nu D_{\mathrm{KL}}\left(\pi\left(s\right),\pi\left(\hat{s}\right)\right)$ for each state s is still computationally expensive. Besides, policy regularization may hinder the expressive power of the policy network, resulting in the sacrifice of optimality and performance. In order to address these issues, we introduce the robust radius of policies and incorporate the Lipschitz continuity into the policy network.

Definition 1. *(Robust radius of policies) Given a stationary policy π, the robust radius of π at state s is defined as the radius of the largest l_∞ ball centered at s, in which π does not change its decision. The formulation is shown as follows:*

$$\mathcal{R}(\pi, s) = \inf_{\pi(s')\neq\pi(s),s'\in\mathcal{S}}\|s' - s\|_\infty. \quad (4)$$

As described in Definition 1, the robust radius of policy π is designed to evaluate policy robustness against observation perturbations quantitatively. We can obtain the following formulation based on Theorem 1:

$$\forall s,\ \mathcal{R}(\pi, s) \ge \epsilon \implies \forall s,\ \min_\nu \widetilde{V}_{\pi\circ\nu}(s) \ge V_\pi(s), \quad (5)$$

which can be proved utilizing Eq. (3) and Eq. (4). The detailed proof is given in Appendix A.2. The Eq. (5) implies that, the policy π can resist all attacks from ν without any degradation in performance when the robust radius is big enough. Therefore, the original problem Eq. (2) can be reformulated as the following equation:

$$\begin{aligned}&\max_{\pi}\mathbb{E}_{s\sim\rho}\left[V_\pi(s)\right]\\ &\text{s.t.}\quad \mathcal{R}(\pi, s) \ge \epsilon, \forall s \in \mathcal{S}.\end{aligned} \quad (6)$$

Note that solving problem Eq. (6) removes the requirement of finding the optimal adversary ν^* compared to Eq. (2).

SortRL Policy Networks

The problem described in Eq. (6) involves computing the robust radius of policy π accurately. However, this task is particularly challenging for typical DNN-based policies due to the high computational cost (Zhai et al. 2019; Zhang et al. 2021a). In this section, we design a novel policy network utilizing the architecture called SortNet (Zhang et al. 2022b) to address this issue with the Lipschitz property.

We utilize a function $g^\pi : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$ to evaluate the score of each action $a \in \mathcal{A}$ based on the perturbed state $\hat{s}$ obtained by the agent. g^π is composed of M-layer fully-connected SortNet (Zhang et al. 2022b). Given a perturbed state s, $\boldsymbol{x}^{(0)} = s$ denotes the input of g^π, and $\boldsymbol{x}_k^{(l)}$ denotes the k-th unit in the l-th layer, which can be computed through the following formulations:

$$\begin{aligned}&x_k^{(l)} = \left(\boldsymbol{w}^{(l,k)}\right)^{\mathrm{T}}\mathrm{sort}\left(\left|\boldsymbol{x}^{(l-1)} + \boldsymbol{b}^{(l,k)}\right|\right),\\ &\omega_i^{(l,k)} = (1-\rho)\rho^{i-1},\ 1 \le l \le M,\ 1 \le k \le d_l,\end{aligned} \quad (7)$$

where d_l is the size of l-th network layer, $\rho \in [0, 1)$ is a hyper-parameter. $\mathrm{sort}(\boldsymbol{x}) \coloneqq \left[x_{[1]}, \cdots, x_{[d]}\right]^{\mathrm{T}}$, where $x_{[k]}$ is the k-th largest element of $\boldsymbol{x} \in \mathbb{R}^d$. The final output $g^\pi(s) = -\left(\boldsymbol{x}^{(M)} + \boldsymbol{b}^{\mathrm{out}}\right)$. Afterward, the agent takes the best action with the highest score:

$$\pi(a|s) \coloneqq \mathbb{1}\left(a = \arg\max_{a_i} g_i^\pi(s)\right), \quad (8)$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. The $\left\{\boldsymbol{b}^{(l,k)}\right\}$ and $\boldsymbol{b}^{\mathrm{out}}$ are network parameters which need to be optimized during training.

Definition 2. *(Lipschitz Continuity) Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, if $\exists K > 0$, such that*

$$\|f(x_1) - f(x_2)\|_p \leq K\|x_1 - x_2\|_p, \forall x_1, x_2 \in \mathbb{R}^n, \quad (9)$$

then f is called K-Lipschitz continuous with respect to l_p norm, where K is the Lipschitz constant. Similarly, a neural network $f : \mathbb{R}^n \to \mathbb{R}^m$ is called l_∞ 1-Lipschitz Neural Network (LNN) if Eq. (9) *holds with $p = +\infty$ and $K = 1$.*

Proposition 1. *The score function $g^\pi(s)$ is* 1*-Lipschitz continuous with respect to l_∞ norm, i.e.*

$$\left\|g^\pi(s_1) - g^\pi(s_2)\right\|_\infty \leq \left\|s_1 - s_2\right\|_\infty, \forall s_1, s_2 \in \mathcal{S}. \quad (10)$$

The detailed proof is given in Appendix A.3.

Theorem 2. *Given a SortRL policy π described in Eq.* (8)*, the lower bound of the robust radius for π can be expressed as follows:*

$$\mathcal{R}(\pi, s) \geq \frac{1}{2}\operatorname{margin}(g^\pi, s), \forall s \in \mathcal{S}, \quad (11)$$

where $\operatorname{margin}(g^\pi, s)$ denotes the difference between the largest and second-largest action scores output by g^π at state s.

The proof of Theorem 2 is given in Appendix A.4. This theorem indicates that, $\forall s \in \mathcal{S}$, if $\operatorname{margin}(g^\pi, s) \geq 2\epsilon$, we can obtain that $\pi(s) = \pi(\hat{s}), \forall \hat{s} \sim \nu(s)$, i.e. the SortRL π can resist attacks from any adversary $\nu \in \mathcal{B}_\epsilon^\infty(s)$. Therefore, the optimization problem described in Eq. (6) can be transformed as follows:

$$\begin{aligned} &\max_\pi \mathbb{E}_{s\sim\rho}\left[V_\pi(s)\right] \\ &\text{s.t.} \quad \pi(a|s) = \mathbb{1}\big(a = \arg\max_{a_i \in \mathcal{A}} g_i^\pi(s)\big), \\ &\qquad \operatorname{margin}(g^\pi, s) \geq 2\epsilon, \forall s \in \mathcal{S}. \end{aligned} \quad (12)$$

Fortunately, the margin defined in Theorem 2 is easy to calculate and can be directly obtained from the network output. Thus, it is practical to improve the robustness of policy π against observation perturbations by optimizing the margin of g^π.

SortRL Training Framework

In this section, we design a training framework for the policy network g^π to solve the problem illustrated in Eq. (12). Different from typical DNNs, the output of each layer in g^π is biased (always being non-negative) under random initialization. The biases of each layer are accumulated, leading to unstable or ineffective outputs of the network, which need to be removed with per-layer normalization, i.e. $\boldsymbol{x}^{(l)} \leftarrow \boldsymbol{x}^{(l)} - \mathbb{E}\left[\boldsymbol{x}^{(l)}\right]$. The estimation of $\mathbb{E}\left[\boldsymbol{x}^{(l)}\right]$ is inaccessible due to the distribution drift of input observations during the training of typical DRL algorithms. More details are given in Appendix B.2.

To address this issue, we introduce a new training pipeline for g^π based on Policy Distillation (PD) (Rusu et al. 2016). Firstly, given a task modeled as $\widetilde{\mathcal{M}}$, a DNN-based teacher policy π_T is trained in the typical MDP $\mathcal{M}$ utilizing arbitrary DRL algorithms, i.e. $\pi_T \leftarrow \arg\max_\pi \mathbb{E}_{s\sim\rho}\left[V_\pi(s)\right]$. An expert dataset $\mathcal{D} \coloneqq \{(s, a^*)\}$ is constructed through interaction between the teacher policy π_T and the clean environment without adversary, where s and a^* denote the clean states and teacher actions correspondingly, i.e. $a^* = \arg\max_a \pi_T(a|s)$.

Afterward, a SortRL policy π_S is constructed as the student policy, which is trained to mimic the decisions of the teacher policy π_T, while maintaining robustness against perturbations. In this work, the π_S is trained by minimizing the following loss function on the expert dataset $\mathcal{D}$:

$$\begin{aligned} \mathcal{L}_{\pi_S} = &\lambda\, \mathbb{E}_{(s,a^*)\sim\mathcal{D}}\left[\mathcal{L}_{\mathrm{CE}}\left(g^\pi(s), a^*\right)\right] \\ &+ \mathbb{E}_{(s,a^*)\sim\mathcal{D}}\left[\mathcal{L}_{\mathrm{Rob}}\left(g^\pi(s), \theta, a^*\right)\right], \end{aligned} \quad (13)$$

where $\lambda \in \mathbb{R}$ is a hyper-parameter. As described in Eq. (13), the $\mathcal{L}_{\mathrm{CE}}(\cdot,\cdot)$ denotes the cross-entropy loss, which is utilized to improve the performance of π_S in the typical MDP $\mathcal{M}$ by mimicking the behaviors of the teacher policy π_T. The formulation of $\mathcal{L}_{\mathrm{CE}}(\cdot,\cdot)$ is described as follows:

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{z}, a^*) = \log\left(\sum_i \mathrm{e}^{z_i}\right) - z_{a^*}, \quad (14)$$

where $\boldsymbol{z} = g^\pi(s)$ denotes the action logits without SoftMax normalization. The $\mathcal{L}_{\mathrm{Rob}}$ utilized in Eq. (13) denotes robustness loss designed based on the Hinge loss. The formulation of $\mathcal{L}_{\mathrm{Rob}}$ is given as follows:

$$\mathcal{L}_{\mathrm{Rob}}(\boldsymbol{z}, \theta, y) = \begin{cases} 0, & z_y < \max\limits_i z_i \text{ or } z_y - \max\limits_{i\neq y} z_i > \theta, \\ \max\limits_{i\neq y} z_i - z_y, & \text{Otherwise}, \end{cases} \quad (15)$$

where $\theta \in \mathbb{R}^+$ is the hinge threshold hyper-parameter. Decisions made by π_S with margins exceeding θ, or deviating from π_T, are excluded from the robustness training. As shown in Eq. (13), the $\mathcal{L}_{\mathrm{Rob}}(\boldsymbol{z}, \theta, y)$ is utilized to improve policy robustness by optimizing the $\operatorname{margin}(g^\pi, s)$ to satisfy the requirement described in Eq. (6) and Eq. (12), i.e. $\mathcal{R}(\pi, s) \geq \frac{1}{2}\operatorname{margin}(\pi, s) \geq \epsilon$.

The parameter λ balances between $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{Rob}}$, corresponding to the trade-off between optimality (nominal performance of π_S in typical $\mathcal{M}$) and robustness (performance against observation perturbations in $\widetilde{\mathcal{M}}$). During the training process, the value of λ is slowly decayed to achieve optimal performance. Initially, we mainly focus on minimizing $\mathcal{L}_{\mathrm{CE}}$ of π_S to learn the decision-making process of π_T. In the later stages, a smaller value of λ is used to prioritize policy robustness against observation perturbations. More details including the pseudocode are given in Appendix B.

Experiment

In this section, to evaluate the performance of our method compared to the existing methods, we conduct experiments on the following three tasks:

a) **Classic Control:** Experiments on four classic control tasks (Brockman et al. 2016) are conducted under different perturbation strength, which aim to demonstrate that SortRL improves the robustness of typical DRL policies.

b) **Video Games:** Afterward, we compare SortRL with existing robust RL methods on six video games against adversarial perturbations with $0 \leq \epsilon \leq \frac{5}{255}$. The purpose is to evaluate the robustness of each method against perturbations on high-dimension observations.

c) **Video Games with Stronger Adversaries:** In order to evaluate the performance of our method against stronger perturbations, we conduct experiments on video games under adversaries with large strength $\epsilon > \frac{5}{255}$, which is quite challenging and rarely studied in previous works (Wu and Vorobeychik 2022).

In this work, all SortRL policies are trained with *AdamW* optimizer (Loshchilov and Hutter 2018) on a single NVIDIA RTX 3090 GPU.

Classic Control

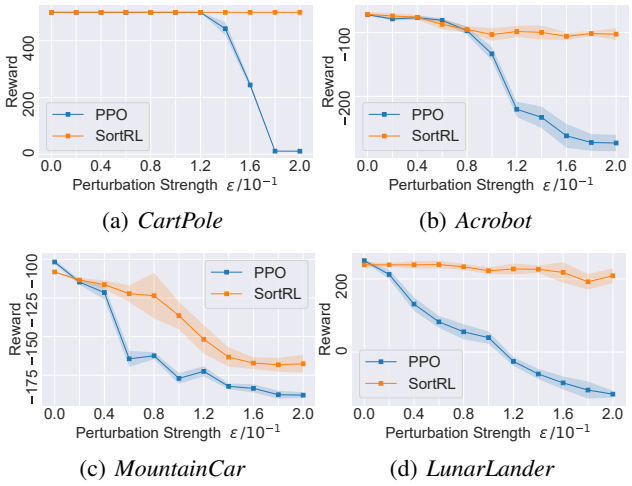
Experimental Settings. In this experiment, four environments are utilized, including *CartPole*, *Acrobot*, *MountainCar*, and *LunarLander*. The policy trained by PPO (Schulman et al. 2017) algorithm is utilized as the teacher π_T. The dataset $\mathcal{D}$ is constructed utilizing π_T with 50K states and corresponding teacher actions. The Projected Gradient Descent (PGD) (Madry et al. 2018) attacker is applied as the adversary ν in this experiment. In each step, the observation is perturbed with untargeted l_∞ PGD attacks with 10 steps. Each method is evaluated with different perturbation strength $\epsilon \in [0.0, 0.2]$, and the episode rewards are recorded to evaluate robustness.

Results and Analysis. The experiment results are given in Fig. 1, where x-axis denotes ϵ value and y-axis denotes episode rewards under perturbations. The mean episode rewards and standard errors are given at ϵ intervals of 0.02, corresponding to curves and shades respectively.

As shown in Fig. 1, our method SortRL (orange) outperforms PPO (blue) with higher rewards generally, especially on tasks with large perturbation strength $\epsilon > 0.1$. The episode rewards of both methods decrease as ϵ increases, but SortRL decays much slower than PPO expert, which demonstrates better robustness of our method. Besides, in some nominal tasks ($\epsilon = 0.0$), there exists a small performance loss of our method compared to PPO, such as *MountainCar* and *LunarLander*. This performance gap is reported and discussed in the previous studies (Liang et al. 2022). One possible explanation is that the robustness loss $\mathcal{L}_{\text{Rob}}$ encourages the policy to become smoother and may harm the expressive power to some extent, which is necessary and crucial for nominal performance.

Video Games

Experimental Settings. In this experiment, we utilize six video games listed as follows. Atari tasks (Bellemare et al. 2013): *Freeway*, *RoadRunner*, *Pong*, and *BankHeist*. ProcGen tasks (Cobbe et al. 2020): *Jumper* and *Coinrun*. Teacher policies are constructed utilizing DQN (Mnih et al. 2015) and PPO (Schulman et al. 2017) for Atari and ProcGen tasks accordingly. The dataset $\mathcal{D}$ is composed of 100k states and corresponding teacher actions. To evaluate the robustness, l_∞-PGD attackers with 10 steps and different strength



(a) *CartPole* (b) *Acrobot*

(c) *MountainCar* (d) *LunarLander*

Figure 1: The experiment results on the classic control tasks.

$\epsilon \in \left\{\frac{1}{255}, \frac{3}{255}, \frac{5}{255}\right\}$ are applied as the adversary ν in this experiment. In each frame, the adversary ν performs untargeted attacks on the input observation, which cheats the policy to change decisions.

Baselines. We compare SortRL with the following representative methods: (1) *Standard DRL algorithms*, including DQN (Mnih et al. 2015), A3C (Mnih et al. 2016), and PPO (Schulman et al. 2017). (2) *RS-DQN* (Fischer et al. 2019) designed with adversarial training and provably robust training. (3) *SA-DQN* (Zhang et al. 2020b) regularizing policy networks based on convex relaxation. (4) *WocalR* (Liang et al. 2022), which estimates and optimizes the worst-case reward of the policy network under bounded attacks. (5) *RADIAL* (Oikarinen et al. 2021), which trains policy networks by adversarial loss functions based on robustness bounds.

Evaluation Metrics. (1) The episode reward against 10 steps PGD perturbations with $\epsilon \in \left\{\frac{1}{255}, \frac{3}{255}, \frac{5}{255}\right\}$, which is widely used in previous works (Fischer et al. 2019; Zhang et al. 2020b; Oikarinen et al. 2021). (2) Action Certification Rate (ACR) (Zhang et al. 2020b), which is designed to evaluate policy performance on certified robustness. ACR is defined as the proportion of the actions during rollout that are guaranteed unchanged with any adversary $\nu \in \mathcal{B}_\epsilon^\infty$. The detailed computation process of ACR for SortRL is given in Appendix D.2.

Results and Analysis. The experiment result are shown in Table 1 (Atari) and Table 2 (ProcGen). As shown in the tables, SortRL outperforms baseline methods and achieves higher episode rewards on video game tasks with different perturbation strength, demonstrating the effectiveness of our approach. Take *RoadRunner* with $\epsilon = 5/255$ as an instance, SortRL achieves an episode reward of 40905 and outperforms existing state-of-the-art 31545 by 29.6%.

As shown in Fig. 2, to measure and analyze the performance, we adopt the metric of average normalized score to aggregate episode rewards across tasks. In detail, given the episode reward Z, its normalized score is defined as

Task	Model/Metric	Episode Reward				ACR (%)
	ϵ	0 (nominal)	$1/255$	$3/255$	$5/255$	$1/255$
Freeway	DQN	33.9 ± 0.07	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	RS-DQN	32.93	32.53	N/A	N/A	N/A
	SA-DQN	30.0 ± 0.0	30.0 ± 0.0	30.05 ± 0.05	27.65 ± 0.22	**100.0**
	WocaR-DQN	31.2 ± 0.4	31.2 ± 0.5	31.4 ± 0.3	21.1 ± 1.75	99.90
	RADIAL-DQN	33.2 ± 0.19	33.35 ± 0.16	33.4 ± 0.13	29.1 ± 0.17	99.82
	SortRL-DQN	$\mathbf{33.91 \pm 0.32}$	$\mathbf{33.83 \pm 0.48}$	$\mathbf{33.94 \pm 0.24}$	$\mathbf{33.92 \pm 0.33}$	99.94
Road Runner	DQN	43390 ± 973	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	A3C	34420 ± 604	31040 ± 2173	3025 ± 317	350 ± 93	0.0
	RS-DQN	12106.67	5753.33	N/A	N/A	N/A
	SA-DQN	$\mathbf{45870 \pm 1380}$	44300 ± 1753	20170 ± 1822	3350 ± 335	60.20
	RADIAL-DQN	44495 ± 1165	44445 ± 1148	39560 ± 1621	23820 ± 942	99.42
	WocaR-DQN	44156 ± 2279	44079 ± 2154	38720 ± 1765	3490 ± 1959	98.41
	RADIAL-A3C	34825 ± 981	31960 ± 933	29920 ± 1496	31545 ± 1480	92.33
	SortRL-DQN	43697 ± 1457	$\mathbf{44596 \pm 1070}$	$\mathbf{39766 \pm 1176}$	$\mathbf{40905 \pm 1249}$	**99.98**
Pong	DQN	$\mathbf{21.0 \pm 0.0}$	-21.0 ± 0.0	-21.0 ± 0.0	-20.85 ± 0.08	0.0
	A3C	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-17.85 ± 0.33	0.0
	RS-DQN	19.73	18.13	N/A	N/A	N/A
	SA-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-19.75 ± 0.1	**100.0**
	WocaR-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	-20.7 ± 0.45	59.05
	RADIAL-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	89.49
	RADIAL-A3C	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	75.53
	SortRL-DQN	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	**100.0**
Bank Heist	DQN	1325.5 ± 5.7	29.5 ± 2.4	0.0 ± 0.0	0.0 ± 0.0	0.0
	A3C	1109.0 ± 21.4	1102.5 ± 49.4	534.5 ± 58.2	115.0 ± 27.8	0.0
	RS-DQN	238.66	190.67	N/A	N/A	N/A
	SA-DQN	1237.6 ± 1.7	1237.0 ± 2.0	1213.0 ± 2.5	1130.0 ± 29.1	97.63
	WocaR-DQN	1220 ± 12	1220 ± 3	1214 ± 7	1094 ± 20	96.75
	RADIAL-DQN	$\mathbf{1349.5 \pm 1.7}$	$\mathbf{1349.5 \pm 1.7}$	$\mathbf{1348 \pm 1.7}$	1182.5 ± 43.3	98.17
	RADIAL-A3C	1036.5 ± 23.4	975 ± 22.2	949 ± 19.5	712 ± 46.4	71.84
	SortRL-DQN	1323.8 ± 6.9	1325.6 ± 6.5	1315.1 ± 5.8	1317.8 ± 7.2	99.69
	SortRL-RADIAL	1342.8 ± 5.5	1340.8 ± 4.7	1345.2 ± 4.9	$\mathbf{1341.1 \pm 5.1}$	**99.93**

Table 1: The experiment results on the Atari video games. The best results are boldfaced, while the second best ones are underlined. The gray rows denote the most robust methods, selected based on the score $R_{\epsilon=0} + \frac{1}{3}\sum_{\epsilon} R_{\epsilon}$, where R_{ϵ} is the mean episode reward given perturbation strength ϵ. N/A denotes the authors have not released results, codes, or models.

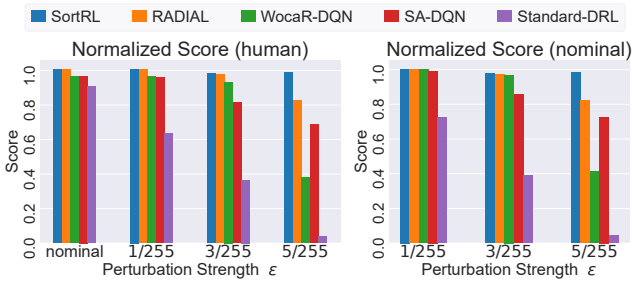


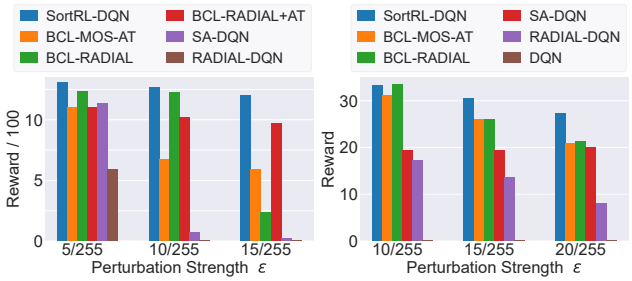
Figure 2: Normalized score on Atari tasks. Left: relative to the human expert. Right: relative to nominal performance.

$\frac{Z-Z_0}{Z_1-Z_0} \in [0,1]$, where Z_0 denotes the reward of the random policy, and Z_1 denotes human reward or nominal reward. As described in Fig. 2, the advantage of SortRL over baseline methods increases as ϵ increases generally. As described in the right figure, compared to the corresponding nominal performance, our method only loses performance less than 1.7% against $\epsilon = 5/255$, while the state-of-the-art RADIAL loses about 18%. Besides, the performance of SortRL on ACR in Table 1 is also excellent, which is greater than 99.6% in various tasks. These results demonstrate that, compared to existing methods, SortRL achieves policy robustness with fewer sacrifices on the optimality and expressive power of the policy network. This relies on the Lipschitz property at the network level and the maximization of the robust radius in the training framework.

It is interesting that SortRL outperforms standard DRL methods in some nominal tasks, such as *Freeway*, *Jumper*, and *Coinrun*. This implies that robust training with suitable parameter settings may improve nominal performance in some tasks. Besides, some methods achieve better performance against perturbations than that in nominal environments, such as in the *Coinrun* task with both *RADIAL* and *SortRL* methods. These interesting phenomena are also observed in previous studies (Zhang et al. 2020b; Oikarinen et al. 2021). One possible explanation is that most tasks pre-

Task	Model/Metric		Episode Reward			
	ϵ	Env. Type	0 (nominal)	$1/255$	$3/255$	$5/255$
Jumper	PPO	Train	8.69 ± 0.11	6.61 ± 0.15	4.50 ± 0.16	3.42 ± 0.15
		Eval	4.22 ± 0.16	3.90 ± 0.15	3.10 ± 0.15	3.15 ± 0.15
	RADIAL-PPO	Train	6.59 ± 0.15	6.70 ± 0.15	6.55 ± 0.15	6.83 ± 0.15
		Eval	3.85 ± 0.15	3.93 ± 0.15	3.75 ± 0.15	3.59 ± 0.15
	SortRL-PPO (Ours)	Train	$\mathbf{9.10 \pm 0.28}$	$\mathbf{9.10 \pm 0.29}$	$\mathbf{9.10 \pm 0.29}$	$\mathbf{9.10 \pm 0.29}$
		Eval	$\mathbf{4.65 \pm 0.39}$	$\mathbf{4.63 \pm 0.39}$	$\mathbf{4.68 \pm 0.39}$	$\mathbf{4.65 \pm 0.39}$
Coinrun	PPO	Train	8.31 ± 0.12	6.36 ± 0.15	4.19 ± 0.16	3.32 ± 0.15
		Eval	6.65 ± 0.15	5.22 ± 0.16	3.58 ± 0.15	3.36 ± 0.15
	RADIAL-PPO	Train	7.12 ± 0.14	7.10 ± 0.14	7.19 ± 0.14	7.34 ± 0.14
		Eval	6.66 ± 0.15	6.71 ± 0.15	6.71 ± 0.15	6.67 ± 0.15
	SortRL-PPO (Ours)	Train	$\mathbf{8.40 \pm 0.37}$	$\mathbf{8.51 \pm 0.36}$	$\mathbf{8.60 \pm 0.35}$	$\mathbf{8.41 \pm 0.37}$
		Eval	$\mathbf{7.33 \pm 0.44}$	$\mathbf{7.70 \pm 0.42}$	$\mathbf{7.23 \pm 0.45}$	$\mathbf{7.20 \pm 0.41}$

Table 2: The experiment results on the ProcGen video games.



(a) *BankHeist* (b) *Freeway*

Figure 3: Experiment results on video games *BankHeist* and *Freeway* with stronger adversaries ($\epsilon \geq 5/255$).

fer smooth policies, i.e. similar decisions given similar observations. However, policies trained by standard DRL are suboptimal due to the non-smoothness property of the policy network, especially in tasks with high-dimension observations, such as ProcGen. Smoother policies and trajectories with higher rewards may be found through robust training or by adding perturbations to the observations.

Video Games with Stronger Adversaries

Experimental Settings In this experiment, the same teacher policies and dataset $\mathcal{D}$ described in video games experiments are utilized. In order to evaluate policy robustness against stronger perturbations, we utilize larger strength $\epsilon > 5/255$. Besides, more attackers (Wu and Vorobeychik 2022) are utilized as adversaries ν in this experiment: (1) PGD attacker with 30 steps. (2) FGSM attacks with Random Initialization (RI-FGSM) (Wong, Rice, and Kolter 2019) (3) RI-FGSM-Multi: sample multiple random starts for RI-FGSM, and choose the first sample which alters the policy decision (4) RI-FGSM-Multi-T: sample multiple random starts for RI-FGSM, and choose the sample which minimizes the estimated Q values among the samples.

Baselines and Evaluation Metrics Most baseline methods in this section are the same as the video games experiments, including (1) *SA-DQN* and (2) *RADIAL*. In addition, a new benchmark (3) *Bootstrapped Opportunistic Adversarial Curriculum Learning (BCL)* (Wu and Vorobeychik 2022) is utilized, which can enhance the robustness of existing robust RL methods under strong adversaries. *BCL* is an adversarial curriculum training framework, and can be combined with various robust RL methods, such as *BCL-RADIAL* and *BCL-MOS-AT*.

Results and Analysis. The experiment results on the *BankHeist* task with $\epsilon \in \left\{\frac{5}{255}, \frac{10}{255}, \frac{15}{255}\right\}$ and *Freeway* task with $\epsilon \in \left\{\frac{10}{255}, \frac{15}{255}, \frac{20}{255}\right\}$ are illustrated in Fig. 3. As shown in the figures, the x-axis denotes the ϵ value while the y-axis denotes episode reward. More experiment results are given in Appendix D.3.

As shown in Fig. 3, SortRL achieves state-of-the-art performance compared to existing methods, especially in tasks with strong perturbations with $\epsilon \geq 15/255$. Take the *Freeway* task with $\epsilon = 20/255$ as an instance, SortRL achieves an episode reward of 27.2 and outperforms existing state-of-the-art BCL-RADIAL (21.2) by approximately 28.3%. The results demonstrate the excellent robustness of SortRL under strong perturbation strength, which relies on the Lipschitz continuity of the policy network and the robust training framework.

Conclusion

In this work, we propose a novel robust RL method called *SortRL*, which improves the robustness of DRL policies against observation perturbations from the perspective of network architecture. We employ a new policy network based on Lipschitz Neural Networks, and provide a convenient approach to optimizing policy robustness based on the output margin. To facilitate training, we design a training framework based on Policy Distillation, which trains the policy to solve given tasks while maintaining a suitable robust radius. Several experiments are conducted to evaluate the robustness of our method, including control tasks and video games with different perturbation strength. The experiment results demonstrate that *SortRL* outperforms existing methods on robustness.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 92248303 and No. 62373242), the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *International conference on machine learning*, 22–31. PMLR.

Afsar, M. M.; Crump, T.; and Far, B. 2022. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7): 1–38.

Anil, C.; Lucas, J.; and Grosse, R. 2019. Sorting out Lipschitz function approximation. In *International Conference on Machine Learning*, 291–301. PMLR.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Cobbe, K.; Hesse, C.; Hilton, J.; and Schulman, J. 2020. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, 2048–2056. PMLR.

Everett, M.; Lütjens, B.; and How, J. P. 2021. Certifiable robustness to adversarial state uncertainty in deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9): 4184–4198.

Eysenbach, B.; and Levine, S. 2021. Maximum Entropy RL (Provably) Solves Some Robust RL Problems. In *International Conference on Learning Representations*.

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

Gouk, H.; Frank, E.; Pfahringer, B.; and Cree, M. J. 2021. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110: 393–416.

Gowal, S.; Dvijotham, K.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Arandjelovic, R.; Mann, T.; and Kohli, P. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.

Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5): 359–366.

Huang, P.; Xu, M.; Fang, F.; and Zhao, D. 2022. Robust reinforcement learning as a stackelberg game via adaptively-regularized adversarial training. *arXiv preprint arXiv:2202.09514*.

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*.

Ju, H.; Juan, R.; Gomez, R.; Nakamura, K.; and Li, G. 2022. Transferring policy of deep reinforcement learning from simulation to reality for robotics. *Nature Machine Intelligence*, 4(12): 1077–1087.

Kaiser, L.; Babaeizadeh, M.; Milos, P.; Osinski, B.; Campbell, R. H.; Czechowski, K.; Erhan, D.; Finn, C.; Kozakowski, P.; Levine, S.; Mohiuddin, A.; Sepassi, R.; Tucker, G.; and Michalewski, H. 2020. Model Based Reinforcement Learning for Atari. In *International Conference on Learning Representations*.

Korkmaz, E. 2021. Investigating vulnerabilities of deep neural policies. In *Uncertainty in Artificial Intelligence*, 1661–1670. PMLR.

Korkmaz, E. 2023. Adversarial Robust Deep Reinforcement Learning Requires Redefining Robustness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 8369–8377.

Kumar, A.; Levine, A.; and Feizi, S. 2022. Policy Smoothing for Provably Robust Reinforcement Learning. In *International Conference on Learning Representations*.

Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; and Hutter, M. 2020. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47): eabc5986.

Li, Q.; Haque, S.; Anil, C.; Lucas, J.; Grosse, R. B.; and Jacobsen, J.-H. 2019. Preventing gradient attenuation in lipschitz constrained convolutional networks. *Advances in neural information processing systems*, 32.

Liang, Y.; Sun, Y.; Zheng, R.; and Huang, F. 2022. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 22547–22561.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

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

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Muratore, F.; Gienger, M.; and Peters, J. 2019. Assessing transferability from simulation to reality for reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 43(4): 1172–1183.

Oikarinen, T.; Zhang, W.; Megretski, A.; Daniel, L.; and Weng, T.-W. 2021. Robust deep reinforcement learning through adversarial loss. *Advances in Neural Information Processing Systems*, 34: 26156–26167.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2040–2042.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2817–2826. PMLR.

Rusu, A. A.; Colmenarejo, S. G.; Gülçehre, Ç.; Desjardins, G.; Kirkpatrick, J.; Pascanu, R.; Mnih, V.; Kavukcuoglu, K.; and Hadsell, R. 2016. Policy Distillation. In Bengio, Y.; and LeCun, Y., eds., *International Conference on Learning Representations*.

Salman, H.; Li, J.; Razenshteyn, I.; Zhang, P.; Zhang, H.; Bubeck, S.; and Yang, G. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32.

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

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*, 8707–8718. PMLR.

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*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In Bengio, Y.; and LeCun, Y., eds., *International Conference on Learning Representations*.

Tessler, C.; Efroni, Y.; and Mannor, S. 2019. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, 6215–6224. PMLR.

Tsuzuku, Y.; Sato, I.; and Sugiyama, M. 2018. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31.

Wang, J.; Liu, Y.; and Li, B. 2020. Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 6202–6209.

Wong, E.; Rice, L.; and Kolter, J. Z. 2019. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.

Wu, F.; Li, L.; Huang, Z.; Vorobeychik, Y.; Zhao, D.; and Li, B. 2022. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. In *International Conference on Learning Representations*.

Wu, J.; and Vorobeychik, Y. 2022. Robust Deep Reinforcement Learning through Bootstrapped Opportunistic Curriculum. In *International Conference on Machine Learning*, 24177–24211. PMLR.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824.

Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; and Kaafar, M. A. 2019. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE vehicular technology magazine*, 14(2): 103–111.

Zhai, R.; Dan, C.; He, D.; Zhang, H.; Gong, B.; Ravikumar, P.; Hsieh, C.-J.; and Wang, L. 2019. MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius. In *International Conference on Learning Representations*.

Zhang, B.; Cai, T.; Lu, Z.; He, D.; and Wang, L. 2021a. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In *International Conference on Machine Learning*, 12368–12379. PMLR.

Zhang, B.; Jiang, D.; He, D.; and Wang, L. 2022a. Boosting the Certified Robustness of L-infinity Distance Nets. In *International Conference on Learning Representations*.

Zhang, B.; Jiang, D.; He, D.; and Wang, L. 2022b. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. *Advances in Neural Information Processing Systems*, 35: 19398–19413.

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

Zhang, H.; Chen, H.; Xiao, C.; Gowal, S.; Stanforth, R.; Li, B.; Boning, D.; and Hsieh, C.-J. 2020a. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *International Conference on Learning Representations*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D.; and Hsieh, C.-J. 2020b. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33: 21024–21037.

Zhao, Y.; Wu, K.; Xu, Z.; Che, Z.; Lu, Q.; Tang, J.; and Liu, C. H. 2022. Cadre: A cascade deep reinforcement learning framework for vision-based autonomous urban driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3481–3489.