

# CAMP in the Odyssey: Provably Robust Reinforcement Learning with Certified Radius Maximization

Derui Wang<sup>♡,♠</sup>, Kristen Moore<sup>♡,♠</sup>, Diksha Goel<sup>♡,♠</sup>, Minjune Kim<sup>♡,♠</sup>, Gang Li<sup>♣</sup>, Yang Li<sup>♣</sup>, Robin Doss<sup>♣</sup>,  
 Minhui Xue<sup>♡,♠</sup>, Bo Li<sup>◇</sup>, Seyit Camtepe<sup>♡,♠</sup>, and Liming Zhu<sup>♡</sup>

<sup>♡</sup>CSIRO's Data61, Australia

<sup>♠</sup>Cyber Security Cooperative Research Centre, Australia

<sup>♣</sup>Deakin University, Australia

<sup>◇</sup>University of Chicago, USA

## Abstract

Deep reinforcement learning (DRL) has gained widespread adoption in control and decision-making tasks due to its strong performance in dynamic environments. However, DRL agents are vulnerable to noisy observations and adversarial attacks, and concerns about the adversarial robustness of DRL systems have emerged. Recent efforts have focused on addressing these robustness issues by establishing rigorous theoretical guarantees for the returns achieved by DRL agents in adversarial settings. Among these approaches, policy smoothing has proven to be an effective and scalable method for certifying the robustness of DRL agents. Nevertheless, existing certifiably robust DRL relies on policies trained with simple Gaussian augmentations, resulting in a suboptimal trade-off between certified robustness and certified return. To address this issue, we introduce a novel paradigm dubbed Certified-radius-Maximizing Policy (CAMP) training. CAMP is designed to enhance DRL policies, achieving better utility without compromising provable robustness. By leveraging the insight that the global certified radius can be derived from local certified radii based on training-time statistics, CAMP formulates a surrogate loss related to the local certified radius and optimizes the policy guided by this surrogate loss. We also introduce *policy imitation* as a novel technique to stabilize CAMP training. Experimental results demonstrate that CAMP significantly improves the robustness-return trade-off across various tasks. Based on the results, CAMP can achieve up to twice the certified expected return compared to that of baselines. Our code is available at <https://github.com/NeuralSec/camp-robust-rl>.

## 1 Introduction

Deep reinforcement learning (DRL) has widespread applications in various areas, including robotics [1, 10, 27], games [38, 50], dialogue systems [32], and finance [62]. However, DRL agents are vulnerable to adversaries who actively perturb the agents' observations. These perturbations cause agents to take sub-optimal actions, potentially leading to severe consequences in critical missions [4, 12, 16, 33, 41, 54].

In robotics applications, attackers manipulating DRL agents's observations can cause permanent damage to the robots or even create life-threatening situations. On the other hand, attacks against DRL in dialogue or trading systems may also lead to biased or harmful decisions by the system. Adversarial attacks against DRL agents typically involve perturbing state observations of the agents or directly modifying their actions. In experimental settings and simulations, the impact of these attacks is often quantified by the sub-optimal rewards obtained by the compromised agents.

Empirical defenses have been developed for DRL agents to mitigate the risk posed by deliberate adversaries. One line of work focuses on training robust DRL policies, aiming to make agents less sensitive to changes in their observations [48, 63, 68]. Methods also exist that model robust Markov Decision Processes (MDPs) to enhance the performance of DRL in noisy or adversarial environments [2, 35, 68]. However, the empirical defenses fail to provide rigid guarantees of the agents' adversarial robustness, leaving them vulnerable to cross-attack generalization and adaptive attacks.

In efforts to verify DRL robustness, randomized smoothing (RS) has emerged as a powerful and scalable tool for both enhancing and certifying the robustness of black-box functions in real-world applications [5]. RS has been applied to DRL policies through policy smoothing, achieving provable robustness against adversarial attacks [23]. Compared to empirical DRL defenses (*e.g.*, adversarial training [20, 57]), RS provides guaranteed lower or upper bounds on the expected return (*i.e.*, expected cumulative discounted rewards going forward) with respect to a certain amount of changes in the observation. Given that smoothing noise from previous steps influences both observations and subsequent smoothing noise, RS in DRL typically constructs a specially structured adversary for certification. This approach enables certification under the Neyman-Pearson lemma or other algorithmic variants [23, 39, 60].

**Motivation and challenges.** A direct motivation for our work is the absence of methods specifically designed to train DRL agents with RS-based provable robustness. Certified robust-

ness through RS often presents challenges that existing methods cannot effectively address. Specifically, there is a trade-off between the certified expected return and the certified radius of adversarial perturbations, leading to two key consequences: 1) First, certifying an expected return against adversaries using large perturbations becomes infeasible, since the certified expected return rapidly drops to zero as the certified radius increases. 2) Additionally, the certified expected return of a randomized agent is significantly lower than the standard return the same agent can achieve in a non-randomized environment, underscoring the need to improve the overall certified expected return. These consequences hinder the deployment of policy smoothing in real-world DRL applications, where both certified robustness and certified utility are crucial.

Existing provable defenses train the policy for smoothing via applying Gaussian noise to the observations [23]. The noise applied during training should match the smoothing noise used in the test phase to prevent degradation in the test-time reward. However, such a policy may not achieve optimal certification results, as the augmentation does not directly maximize the certified radius during training, leaving the correlation between the augmentation and the trade-off unclear. On the other hand, there also exist robust DRL training methods that enhance the robustness and smoothness of DRL agents operating in noisy or adversarial environments [35, 48, 63, 68]. However, instead of improving the certified expectations of returns received through randomized policies, these methods primarily focus on the epistemic robustness of deterministic policies. Moreover, the desired goals of robustness and utility cannot be easily achieved using existing methods for training classifiers with certified robustness [18, 19, 46, 65]. The reasons for this are 1) DRL tasks are MDPs, which introduce optimization objectives that diverge from that of classification tasks. 2) The certified radius of perturbations is determined by numerical methods like binary search, which makes it difficult to directly optimize the radius in order to enhance the robustness. These two key characteristics of DRL necessitate novel paradigms in the construction and training of the policy.

**Our stance and contributions.** Motivated by these factors, we propose Certified-rAdius-Maximizing Policy (CAMP) training, a method for developing DRL agents that achieve better certified expected returns without compromising certified robustness. Theoretically, we derive a substitute certified radius that can be maximized in the training, in parallel to the utility maximization process. This enables CAMP to mitigate the trade-off between the certified expected return and the certified radius by offering a superior base function. Our focus is on agents trained using deep Q-learning [38] in environments with discrete action spaces. Through a change of variables, we formally define the certified radius as a function of a target threshold for the certified expected return. This certified radius is then transformed into a differentiable format for further analysis. Building on the insight that the decline in optimal

certified expected return is due to accumulated per-step errors, we derive a local certified radius that serves as a surrogate loss for optimizing the certified radius.

Furthermore, to enhance the versatility and training stability of CAMP, we introduce a novel training paradigm called policy imitation. Since the application of CAMP can influence the Bellman error, it may result in the overestimation of Q-values during training. This issue can impede the convergence of the training process and often leads to suboptimal policies [42, 56]. Previous solutions, such as Double DQN [56], ensemble bootstrapping [42], and conservative smoothing [48, 63], are not designed to directly address this problem within the CAMP framework. Consequently, a new training paradigm is needed to effectively integrate CAMP into DRL training pipelines. Importantly, we aim for CAMP to be applicable across various learning environments. Policy imitation addresses the aforementioned challenges by using a reference network to model the oracle Q-values, which can then be used to constrain the primary policy during training. The reference network is trained using conventional temporal-difference methods, while the primary network imitates the reference network, minimizing the difference between their predicted actions for the same observations. Additionally, the CAMP loss is applied to the primary policy to enhance its certified robustness radius. To summarize, our contributions are as follows:

- We analyze the certified radius for globally smoothed policies and reformulate it as a radius maximization objective, which can be directly optimized during the training process.
- We propose a novel policy imitation training paradigm that integrates certified radius maximization into deep Q-learning.
- CAMP achieves higher certified expected returns at larger certified radii and demonstrates superior empirical robustness in individual game episodes.

In the following paper, we will first introduce the necessary background knowledge, along with the definitions of the problem and threat model. We will then proceed to present CAMP and discuss the associated experiments.

## 2 Preliminaries

To begin, we provide a brief overview of deep Q-learning, adversarial attacks on DRL, and the certified robustness of DRL. This paper focuses on discrete action spaces and emphasizes robustness certification using policy smoothing and the Neyman-Pearson Lemma, as these methods offer a tight bound on the certified expected return.

### 2.1 Deep Q-Learning

A reinforcement learning task is usually defined as a discrete MDP. Specifically, an agent interacts with an environment  $E = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, s_0)$ , where  $\mathcal{S} \in \mathbb{R}^d$  and  $\mathcal{A} \in \mathbb{R}^k$  are the

space of states and the space of actions, respectively.  $\mathcal{T}(\cdot)$  is a state transition function and  $\mathcal{R}(\cdot)$  is a reward function.  $s_0$  comes from an initial distribution of states.  $\gamma \in [0, 1]$  is a discount factor for the reward at different steps. Furthermore, the state at the  $t+1$ -th step is determined by  $s_{t+1} = \mathcal{T}(s_t, a_t)$ . In this paper, following previous certification settings, the states are fully observable by a deterministic observation function. Therefore, we use  $s$  interchangeably to denote both the state and the observation. Given the state  $s_t$  and the corresponding action  $a_t$  at the  $t$ -th step, the reward  $r_{t+1}$  acquired by  $a_t$  is computed by the reward function as  $r_{t+1} = \mathcal{R}(s_t, a_t)$ .

In practice, the transition function is determined by nature and is therefore unknown, which encourages the development of model-free RL algorithms. Among these algorithms, Q-learning [59] aims to model the action-value function (*i.e.*, Q-function)  $Q_\pi(s, a)$  which estimates the expected future return for taking action  $a$  in state  $s$  and following a policy. DQN models  $Q_\pi(s, a)$  with a neural network (Q-network) and uses a greedy policy to select actions. For simplicity, we use  $\pi$  to denote both the Q-network and its parameters in this paper. Thus, at the  $t$  step, the action selection policy can be formulated as taking an observed state  $s_t$ , predicting a probability simplex  $\{\pi(s_t)_i\}_{i=1}^{|\mathbb{A}|}$  over  $\mathbb{A}$ , and then selecting  $a = \arg \max_i \pi(s_t)_i$ .

To update the network parameters  $\pi$  towards an optimal Q-function, a Temporal-Difference (TD) loss is defined as the difference between the current Q-value and an estimated optimal value. Specifically, the optimal Q-value is the solution to the Bellman equation, which can be calculated as:

$$Q_{\pi^*}(s, a) = \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} \left[ \max_{a^*} Q_{\pi^*}(s', a^*) \right], \quad (1)$$

where  $\gamma \in [0, 1]$  is a discount factor penalizing the rewards in the future.  $Q_{\pi^*}(s', a^*)$  is the optimal Q-value in which  $s'$  is the next state according to the transition function  $\mathcal{T}(s, a)$  and  $a^*$  is the best action estimated greedily for the next step by the optimal network denoted as  $\pi^*$ . The overall training objective of the agent can be formulated as minimizing the mean squared Bellman error between the current Q-function and the optimal Q-function:

$$\min_{\pi} \mathbb{E}_{(s, a, s') \sim \mathbf{Z}} \left[ Q_\pi(s, a) - \left( \mathcal{R}(s, a) + \gamma \max_{a'} Q_{\pi^*}(s', a') \right) \right]^2, \quad (2)$$

where  $\mathbf{Z}$  is the space of state-action trajectories. In practice, an experience replay buffer or an offline dataset stores a set of previously collected trajectories as samples from  $\mathbf{Z}$ , such that the above training objective can be solved by empirical risk minimization.

## 2.2 Expected Return Certification against Perturbed Observations

**Adversarial observation perturbations against DRL.** An adversary targeting a DRL agent aims to perturb the agent's

observations during test time, thereby minimizing the return within a finite horizon  $T$ . A simple adversarial attack adapts the Fast Gradient Sign Method (FGSM) to perturb observations with a sequence of perturbations  $\Delta = (\delta_0, \delta_1, \dots, \delta_{T-1})$ ,  $\Delta \in \mathbb{R}^{d \times T}$  and alter the actions of the DRL agent, causing the agent to reject the correct action and instead select the one with the minimal Q-value [16]. On the other hand, another type of attack finds a perturbation sequence  $\Delta$ , as the solution of

$$\arg \min_{\Delta} \mathbb{E}_{\delta_t \in \Delta, s_t} \left[ \sum_{t=0}^{T-1} \gamma^t \mathcal{R} \left( s_t, \arg \max_a \pi(s_t + \delta_t)_a \right) \right], \quad (3)$$

In practice, the attacker can find per-state perturbation as  $\delta_t = \arg \min_{\delta_t} \mathcal{D}_J(\pi(s_t) \| \pi(s_t + \delta_t))$ , where  $\mathcal{D}_J$  is the Jeffrey's Divergence [63]. In both attacks, the adversarial perturbations in the sequence  $\Delta$  are collectively constrained within a region  $B(\Delta) = \{(\delta_0, \delta_1, \dots, \delta_{T-1}) : \sqrt{\sum_{t=0}^{T-1} \|\delta_t\|_2^2} \leq \tau\}$  under a budget of  $\tau$ .

**Robustness certification via policy smoothing.** In response, policy smoothing emerges as a provable defense against adversarial attacks. Policy smoothing involves adding noise sampled from a Gaussian distribution  $\mathcal{N}(0, \sigma^2 I)$  with variance  $\sigma^2$  to state observations at each step, thereby randomizing the state-action trajectory of the DRL agent. By repeating this process across multiple runs, a set of randomized trajectories along with their corresponding rewards can be collected, allowing the expected return of the agent to be lower bounded.

Specifically, let the sequence of the smoothing noises across  $T$  steps (*e.g.*, within an episode) be  $\varepsilon = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{T-1})$ , the defender can sample random trajectories  $\mathbf{z} = (s_0 + \varepsilon_0, a_0, \dots, s_{T-1} + \varepsilon_{T-1}, a_{T-1})$ ,  $\mathbf{z} \in \mathbf{Z}$ , where  $\mathbf{Z}$  is the space of randomized state-action trajectories. Furthermore, a randomized reward  $r_{t+1} = \mathcal{R}(s_t, \arg \max_a \pi(s_t + \varepsilon_t)_a)$  is received given the observed state and action output by the policy. The return obtained by a random trajectory is  $\mathbf{r} = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$  and  $\gamma$  is the discount factor. Therefore, such a return can be viewed as a randomized return function  $\mathbf{r} = F_\pi(\mathbf{z})$  of the trajectory  $\mathbf{z}$ , parameterized by the policy parameters  $\pi$ . Let us consider the probability  $P_\pi^{\mathbf{z}}(C) = \Pr[F_\pi(\mathbf{z}) \geq C]$ , and we can accordingly define  $\Psi(C) = 1 - P_\pi^{\mathbf{z}}(C)$ . It is easy to notice that  $\Psi(C)$  is a cumulative distribution function (CDF), *i.e.*,  $\Psi(C) = \Pr[F_\pi(\mathbf{z}) \leq C]$ . By sampling  $F_\pi(\mathbf{z})$  values for  $m$  times via allowing the agent to play on random trajectories for  $m$  runs, a set  $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$  of return values sorted in ascending order can be obtained to compute an empirical cumulative distribution function (ECDF)  $\tilde{\Psi}(C) = \sum_{i=1}^m \mathbb{1}_{\{\mathbf{r}_i \leq C\}} / m, \forall \mathbf{r}_i \in \mathbf{R}$ . Next, the interval  $[\underline{\Psi}(C), \bar{\Psi}(C)]$  of the CDF  $\Psi(C)$  can be computed from the ECDF by Dvoretzky–Kiefer–Wolfowitz Inequality [9]:

$$\bar{\Psi}(C) = \tilde{\Psi}(C) + \sqrt{\frac{\ln(2/\alpha)}{2m}}, \quad (4)$$

with probability  $1 - \alpha$  and the draw count  $m$ . Since  $\Psi(C) = 1 - P_\pi^{\mathbf{z}}(C)$ , the upper bound  $\bar{\Psi}(C)$  can thus be used to obtain

a probabilistic lower bound of  $P_\pi^z(C)$ , which gives  $\underline{P}_\pi^z(C) = 1 - \overline{\Psi}(C)$ .

Consider an adversary who adds a perturbation sequence  $\Delta = (\delta_1, \delta_2, \dots, \delta_{T-1})$  under a  $\ell_2$ -norm budget of  $\tau$  (i.e.,  $\|\Delta\|_2 = \sqrt{\sum_{t=0}^{T-1} \|\delta_t\|_2^2}$ ,  $\|\Delta\|_2 \leq \tau$ ) to the state observations and let the perturbed random trajectory be  $\mathbf{z}' := \mathbf{z} + \Delta = (s'_0 + \varepsilon_0, a'_0, \dots, s'_{T-1} + \varepsilon_{T-1}, a'_{T-1})$  in which  $s'_t = s_t + \delta_t$  and  $a'_t = \arg \max_a \pi(s_t + \varepsilon_t + \delta_t)_a$ . The following perturbation bound on the probability  $P_\pi^z(C)$  can be established by an adaptive Neyman-Pearson Lemma based on a structured adversary [23]:

$$P_\pi^z(C) \geq \Phi\left(\Phi^{-1}\left(\underline{P}_\pi^z(C)\right) - \frac{\tau}{\sigma}\right), \forall \|\Delta\|_2 \leq \tau. \quad (5)$$

Furthermore, since the expectation of  $F_\pi(\mathbf{z})$  can be represented as the integral of  $P_\pi^z(C)$  over  $C$  values [24], a lower bound on the expected return under adversarial perturbations  $\Delta$  can be obtained, given  $C \in \mathbf{R}$ , as:

$$\begin{aligned} \mathbb{E}[F_\pi(\mathbf{z}')] &\geq \mathbf{r}_1 \cdot \Phi\left(\Phi^{-1}\left(\underline{P}_\pi^z(\mathbf{r}_1)\right) - \frac{\tau}{\sigma}\right) \\ &+ \sum_{i=2}^m (\mathbf{r}_i - \mathbf{r}_{i-1}) \cdot \Phi\left(\Phi^{-1}\left(\underline{P}_\pi^z(\mathbf{r}_i)\right) - \frac{\tau}{\sigma}\right), \quad (6) \\ &\forall \|\Delta\|_2 \leq \tau. \end{aligned}$$

On the right-hand side,  $\Phi(\cdot)$  is the cumulative density function (CDF) of standard Gaussian and  $\Phi^{-1}(\cdot)$  is its inverse.

### 3 Problem Formulation

In this section, we present precise formulations of the research problem. We also define the threat model to reflect the settings for training and certifying DRL agents, as well as the key properties of the adversary.

#### 3.1 Problem Statement

We aim to maximize both the expected return and the robustness of the DRL agent. This desideratum can be decomposed into two sub-objectives: minimizing utility cost and minimizing robustness cost. Moreover, both cost minimizations should be seamlessly integrated into the training process of the DRL agent.

Following the above notations and Equations, we can observe that the lower bound of the expected return in Equation 6 has a positive correlation with the probabilities in the set  $\{\underline{P}_\pi^z(\mathbf{r}_i)\}_{i=1}^m$ . This observation implies that there is a monotonic connection between the utility and the probability of obtaining a return above each threshold  $C_i$ . We can thus instantly formulate a utility cost as:

$$\ell_{ut}(\pi) = \sum_{i=1}^m \left(1 - \underline{P}_\pi^z(\mathbf{r}_i)\right). \quad (7)$$

However, this formulation makes minimizing  $\ell_{ut}(\pi)$  an intractable problem due to non-differentiable  $\underline{P}_\pi^z(\mathbf{r}_i)$  and the sampling of  $\mathbf{z}$  and  $\mathbf{R}$ . To optimize  $\ell_{ut}(\cdot)$ , we will construct a differentiable surrogate loss in Section 4 as an upper bound of Equation 7.

On the side of robustness, we also aim to derive a loss function indicating the extent of robustness. It is natural to consider a closed-form representation of the certified radius as a function of  $\pi$  to measure the robustness. However, in policy smoothing, the certified radius (i.e., the maximal  $\tau$  making Equation 6 hold) is a variable rather than a function and is usually determined by binary search. To address this challenge, it is noticeable that the certified lower bound in Equation 6 monotonically decreases with increasing  $\tau$ . Therefore, the binary search can be viewed as finding a  $\tau$  value such that the lower bound is no less than a predetermined value  $\xi$ . To clearly formulate the process of optimizing  $\tau$ , our analysis stems from the following theorem:

**Theorem 1 (Change of variable).** *Given a target expected return threshold  $\xi$  for the randomized policy, let the perturbed trajectory be  $\mathbf{z}' = \mathbf{z} + \Delta$  and define  $P_\pi^z(C) := \Pr[F_\pi(\mathbf{z}) \geq C]$ . Let  $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$  represent a set of sampled and sorted values of  $F_\pi(\mathbf{z})$  such that  $\mathbf{r}_1 \leq \mathbf{r}_2 \leq \dots \leq \mathbf{r}_m$ . If  $\xi/\mathbf{r}_1 \leq \underline{P}_\pi^z(\mathbf{r}_1)$  and*

$$\|\Delta\|_2 \leq \sigma \left[ \Phi^{-1}(\underline{P}_\pi^z(\mathbf{r}_1)) - \Phi^{-1}(\xi/\mathbf{r}_1) \right], \quad (8)$$

*then  $\mathbb{E}[F_\pi(\mathbf{z}')] \geq \xi$ .*

The proof of the theorem is in Appendix A. Theorem 1 implies that  $\tau = \sigma[\Phi^{-1}(\underline{P}_\pi^z(\mathbf{r}_1)) - \Phi^{-1}(\xi/\mathbf{r}_1)]$ , thereby converting  $\tau$  from a variable to a function of  $\xi$ . In this way, it defines a substitute certified radius and characterizes the relationship between this radius, the Q-network  $\pi$ , the statistics  $\underline{P}_\pi^z(\mathbf{r}_1)$ , and  $\xi$ . However, the substitute certified radius remains non-differentiable at this stage. Therefore, in Section 4, we propose a method to maximize the certified robustness radius by optimizing each local certified radius over a finite horizon  $T$ .

#### 3.2 Threat Model

CAMP operates during the training stage of DRL agents. The defender, responsible for both model training and robustness certification, follows the typical practices of DRL trainers. The defender must have access to the observations of the DRL agent and can modify these observations with random noise or perturbations. This is usually feasible, as the defender typically sets up the environment in which the DRL agent is trained. Additionally, as the model trainer, the defender has full access to the Q-network and the policy, including intermediate weights and predicted Q values based on observations. For certification purposes, the defender can test the trained agent in various environments over multiple rounds and record the actual returns from each round.

The adversaries against which we certify our defense are those who perturb the observed states of the DRL agent to



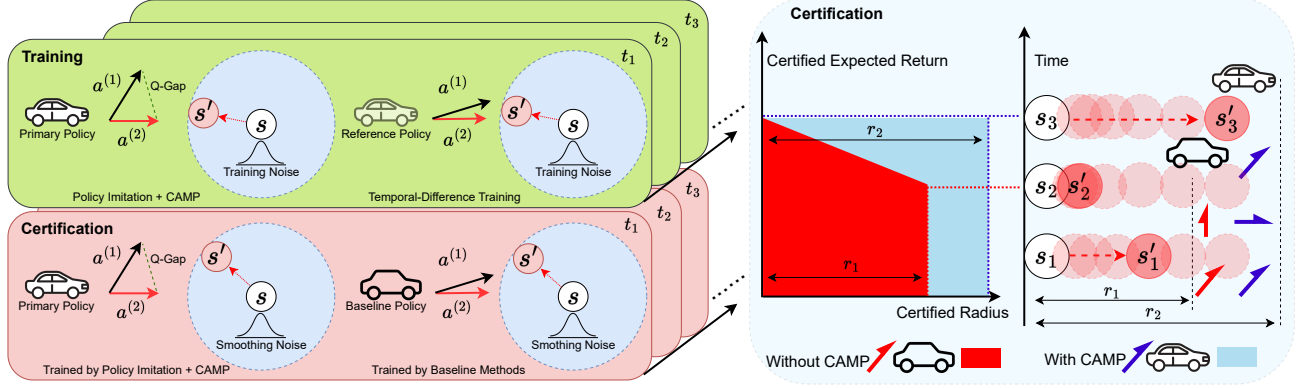


Figure 1: An overview of CAMP. During training, the agents interact with the environments with observation noise and receive rewards. A reference policy has a reference Q-network learning through a vanilla temporal-difference loss while the Q-network of a primary policy is optimized by minimizing the CAMP loss to increase the gap between the top-1 and runner-up Q-values. The primary network also mimics the action predicted by the reference network during training with the imitation loss. The trained primary policy can then be certified by policy smoothing with better certified expected return at each certified radius.

mislead its actions. For instance, in classic control applications, such as robotic arm manipulation, the observed states may consist of sensor data related to the arm’s kinematics. An attacker could manipulate this sensor data to induce abnormal movements in the robotic arm, while the underlying state used for transition and reward computation remains unchanged. This observation-based adversary is more practical than attackers perturbing actions or policy parameters in real-world scenarios, as it is generally easier to tamper with sensor data than to manipulate agent actions or alter parameters in real time. Furthermore, this type of adversary presents significant challenges for certified robustness due to the non-convex and black-box nature of policy functions, which motivates the adoption of RS for certification.

Additionally, adversaries can distribute perturbations across a finite number of transitions in the environment, provided that the total  $\ell_2$  norm of the perturbations remains within a specified budget. This setup allows adversaries to flexibly distribute perturbations, leading to stronger attacks, measurable by reduced DRL agent returns. The adversary can access the victim agent’s Q-network in either a white-box or black-box manner, allowing them to optimize perturbations more effectively. In white-box attacks, adversaries have access to the network’s parameters, states, and actions, while black-box attacks rely on querying the Q-network with observed states to infer the corresponding actions.

### 3.3 Design Intuition

In this paper, we propose CAMP as a method for training provably robust DRL agents in discrete action spaces through certified radius maximization. CAMP tackles this challenge by formulating the robustness radius and transforming it into a robustness cost that can be minimized in the deep Q-learning

process. To start with, we first convert the certified radius into a function of the Q-network  $\pi$ , the statistics  $P_{\pi}^z(\mathbf{r}_1)$ , and a threshold  $\xi$ . Building on the Lipschitz continuity of the expected return function with respect to perturbations in the observations, we proceed to derive a soft radius. In addition, the observation noise perturbs the agent’s action selection, thereby affecting the action-value function at each decision step. A robust policy is one that consistently selects the action yielding the highest expected Q-value in the presence of observation noise at every step. From this principle, we derive local radii that exhibit a positive correlation with the gap between the top-1 and runner-up Q-values.

Importantly, we find that the robustness cost cannot be minimized with the TD loss at the same time, as it destabilizes the training process. To overcome this barrier, we propose adopting policy imitation as a training paradigm. Policy imitation creates a policy based on a reference network to guide the training of the Q-network of the primary policy. The primary network mimics the action selected by the reference network but does not directly approach the Q-value of the state-action pair. On the other hand, the primary policy minimizes its robustness cost during the imitation, such that agents with this trained primary policy can be certified at a greater robustness radius given a fixed certified expected return. The detailed design of CAMP will be introduced in the next section.

## 4 Provable Robustness via CAMP

This section introduces the design of CAMP. The core objective of CAMP is to train the agent to maximize the certified utility (e.g., achieve a higher certified expected return) while optimizing an approximated, differentiable certified radius. We first describe the steps leading to the differentiable local certified radius and demonstrate how maximizing this radius improves

certification performance. Next, to prevent overestimation of Q-values during the training of CAMP, we introduce *policy imitation* as a novel training paradigm. Policy imitation employs a secondary Q-network as an oracle to guide the training of the primary policy. Finally, we certify the expected returns of the CAMP agent through policy smoothing. An overview of the CAMP pipeline is provided in Figure 1.

## 4.1 Expected Return Maximization

Our first desideratum is to maximize the certified utility of the certifiably robust DRL agent. In reference to the objective outlined in Section 3.1, our goal is to find a differentiable upper bound for Equation 7. The initial challenge is to address the sampling process of  $\mathbf{z}$  and  $\mathbf{R}$ . Intuitively, increasing the probability that the return from random trajectories exceeds a certain threshold can be equated to maximizing the expectation of returns over randomness in some special cases. First, we restate the following well-known theorem:

**Theorem 2** (*Correlation between CDF and expectation*). *Given two random variables  $X \geq 0$  and  $Y \geq 0$ , let their expectations be  $\mathbb{E}[X]$  and  $\mathbb{E}[Y]$ , respectively. The following inequality holds if and only if  $\mathbb{E}[X] \leq \mathbb{E}[Y]$ :*

$$\int_0^{+\infty} [\Psi_X(C) - \Psi_Y(C)] dC \geq 0, \quad (9)$$

where  $\Psi_X(C)$  and  $\Psi_Y(C)$  are the CDFs of  $X$  and  $Y$ , respectively.

The theorem suggests that increasing the expectation of a random variable may increase the chance of the random variable exceeding a given threshold, thereby decreasing the integral of the CDF. In our case, given the random variable  $F_\pi(\mathbf{z})$ , we can increase its expected value such that Equation 7 can be reduced, as it is positively correlated with the CDF. This motivates us to formulate the utility maximization objective as minimizing the expected TD loss,  $\ell_{ut}(\pi; s, \varepsilon, s', \varepsilon')$ , over randomized trajectories.

$$\min_{\pi} \mathbb{E}_{(s, \varepsilon, s', \varepsilon') \sim \mathcal{Z}} [\ell_{ut}(\pi; s, \varepsilon, s', \varepsilon')]. \quad (10)$$

In the case of DQN,  $\ell_{ut}(\cdot)$  is the mean squared Bellman error with noisy observations, as analogous to Equation 2. Therefore, the utility loss is defined as:

$$\ell_{ut}(\pi; s, \varepsilon, s', \varepsilon') = \left[ Q_\pi(s + \varepsilon, a) - \left( \mathcal{R}(s, a) + \gamma \max_{a'} Q_{\pi^*}(s' + \varepsilon', a') \right) \right]^2, \quad (11)$$

where  $\varepsilon, \varepsilon' \sim \mathcal{N}(0, \sigma^2 I)$ ,  $a = \arg \max_i \pi(s + \varepsilon)_i$ , and  $\pi^*$  is the targeted optimal Q-network as defined in Section 2.1.

Unlike Equation 2, the error here is computed not only over the state observations and actions, but also over the smoothing noise draws  $\varepsilon$  and  $\varepsilon'$ . This loss function actually recovers the Gaussian-augmented training of DRL agents in the previous

papers [23, 60]. However, it does not directly address the robustness of the agent. Therefore, we will proceed to the next sections to explain how the certified radius can be enhanced by constructing specialized training objectives.

## 4.2 Certified Radius Maximization

According to the problem definition in Section 3.1, the crux of the radius maximization involves two key aspects. First, we need to build the dependence between the certified radius and the Q-network parameters  $\pi$ . Second, a differentiable loss of the certified radius with respect to  $\pi$  should be devised.

We begin with a lemma, which is a generalization from Lemma 2 in SmoothAdv [46] and further proved in Lemma 2 of CROP [60]. The lemma is restated as follows.

**Lemma 1** (*Lipschitz continuity of smoothed return function*). *For any measurable function  $F_\pi : \mathbf{z} \in (\mathbb{S} \times \mathbb{A} \times \mathbb{R}^d)^T \rightarrow [A, B]$ , let  $\mathbb{E}[F_\pi(\mathbf{z})] = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} F_\pi(\mathbf{z} + \varepsilon)$ . There is  $\mathbf{z} \rightarrow \mathbb{E}[F_\pi(\mathbf{z})]$  is  $\frac{(B-A)}{\sigma} \sqrt{2/\pi}$ -Lipschitz.*

The Lipschitz continuity immediately implies that  $\frac{\sigma}{(B-A)\sqrt{2/\pi}} (\mathbb{E}[F_\pi(\mathbf{z})] - \xi)$  serves as a certified radius, albeit a loosely certified one. Although it does not directly bound the radius defined in Equation 8, this soft radius is positively correlated with the previous radius. Specifically, when  $\mathbf{r}_1$  is fixed, both radii monotonically increase with  $\mathbb{E}[F_\pi(\mathbf{z})] - \xi$ . Fortunately, this implication also suggests that this soft radius can be used as a surrogate for optimization. Building on the above lemma, we now proceed to the main theorem of deriving a differentiable soft certified radius.

**Theorem 3** (*Soft certified radius*). *Given a target expected return  $\xi \leq \mathbb{E}[F_\pi(\mathbf{z})]$  where  $F_\pi : \mathbf{z} \in (\mathbb{S} \times \mathbb{A} \times \mathbb{R}^d)^T \rightarrow [A, B]$ , suppose an adversary perturbs the observed states by applying  $\Delta = (\delta_0, \delta_1, \dots, \delta_{T-1})$ . The target expected return will not drop below  $\xi$  if the perturbations satisfy:*

$$\|\Delta\|_2 \leq \sigma \left[ \Phi^{-1} \left( \frac{\mathbb{E}[F_\pi(\mathbf{z})] - A}{B - A} \right) - \Phi^{-1} \left( \frac{\xi - A}{B - A} \right) \right]. \quad (12)$$

*Proof.* Based on Lemma 1, the mapping  $\mathbf{z} \rightarrow \frac{\mathbb{E}[F_\pi(\mathbf{z})] - A}{B - A}$  is  $\sqrt{(2/\pi)}/\sigma$ -Lipschitz. Moreover, based on the same proof of Lemma 2 of Salman et al. [46],  $\Phi^{-1}(\frac{\mathbb{E}[F_\pi(\mathbf{z})] - A}{B - A})$  is  $\frac{1}{\sigma}$ -Lipschitz. We have

$$\sigma \left[ \Phi^{-1} \left( \frac{\mathbb{E}[F_\pi(\mathbf{z})] - A}{B - A} \right) - \Phi^{-1} \left( \frac{\mathbb{E}[F_\pi(\mathbf{z} + \Delta)] - A}{B - A} \right) \right] \leq \|\Delta\|_2. \quad (13)$$

Thus, if the perturbation  $\Delta$  satisfies:

$$\|\Delta\|_2 \leq \sigma \left[ \Phi^{-1} \left( \frac{\mathbb{E}[F_\pi(\mathbf{z})] - A}{B - A} \right) - \Phi^{-1} \left( \frac{\xi - A}{B - A} \right) \right], \quad (14)$$

by the monotonicity of  $\Phi^{-1}(\cdot)$ , we obtain

$$\mathbb{E}[F_\pi(\mathbf{z} + \Delta)] \geq \xi. \quad (15)$$

□

This soft radius is computed through the expected return over clean trajectories, which means it can be efficiently sampled and calculated as a global radius for the  $T$ -step MDP. Since Theorem 3 characterizes the globally certified radius for a trajectory of length  $T$ , it is necessary to decompose the global radius into local certified radii obtained at each time  $t$ .

The global certified radius can be decomposed into per-step action selection errors, which can, in turn, be bounded by a local robustness radius. Specifically, since the radius positively correlates with  $\mathbb{E}[F_\pi(\mathbf{z})] - \xi$ , given a fixed  $\xi$ , maximizing the soft radius is equivalent to increasing the expected return over randomized state-action trajectories. It is important to note that for discrete action spaces, if the optimal greedy policy is derived from a Q-network with parameters  $\pi^*$ , it is expected to select the action  $a_t$  that maximizes the expectation of the action-value function at each time  $t$ . In this context, the probability of a perturbation at step  $t$  reducing the optimal return is equivalent to the probability that the perturbation causes the predicted action to shift from correct to incorrect. We can thus formulate the optimal expected action-value function based on this optimal policy as  $\bar{Q}_{\pi^*}(s_t, a_t) = \max_{\pi} \mathbb{E}_{\epsilon_t} Q_\pi(s_t + \epsilon_t, a_t)$ . Next, we have the following theorem:

**Theorem 4 (Local certified radius).** *Let  $l_i$  and  $u_i$  denote the lower and upper bounds of the expected action-value function at step  $i$ , respectively. Let the perturbations from step  $t$  to step  $T-1$  be  $\{\delta_i\}_{i=t}^{T-1}$ . Under a greedy policy, the optimal reward at step  $t$  is obtained by taking action  $a_i^{(1)} = \arg \max_{a_i} \bar{Q}_{\pi^*}(s_i, a_i)$ . The optimal expected return from step  $t$  to  $T-1$  will not be reduced if the local perturbation  $\delta_i$  at each step  $i$  satisfies:*

$$\|\delta_i\|_2 \leq \frac{\sigma}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(2)}) - l_i}{u_i - l_i} \right) \right], \quad (16)$$

where  $a_i^{(2)}$  is the runner-up action  $a_i^{(2)} = \arg \max_{a_i: a_i \neq a_i^{(1)}} \bar{Q}_{\pi^*}(s_i, a_i)$ .

*Proof sketch.* The proof stems from the Lipschitz continuity of  $\bar{Q}_{\pi^*}(s_t, a_t)$  with respect to perturbations in the observed states. Since  $\bar{Q}_{\pi^*}(s_t, a_t)$  is Lipschitz continuous, by comparing the  $\delta_t$ -perturbed  $\bar{Q}_{\pi^*}(s_t + \delta_t, a_t)$  with the unperturbed  $\bar{Q}_{\pi^*}(s_t, a_t)$  for different actions, and using the monotonicity of  $\Phi^{-1}(\cdot)$ , we can determine the  $\ell_2$  norm of the maximum allowable perturbation such that the action inducing the maximum reward remains selected by the greedy policy at each step  $t$ .  $\square$

This  $\ell_2$  norm represents a local radius within which the perturbation does not compromise the optimal expected Q-value obtained by the optimal policy. Preserving the optimal expected return can therefore be converted to a problem of

increasing the gap between the top-1 and runner-up Q-values under the optimal policy. We refer to this gap as the Q-gap in subsequent discussions. The full proof of the theorem is in Appendix A. The local certified radius at each step can be obtained by iteratively applying Theorem 4 from  $t = 0$  to  $t = T-1$ . Another implication of Theorem 4 is that the overall robustness budget  $\mathbb{E}[F_\pi(\mathbf{z})] - \xi$  can be distributed across per-step budgets, reflecting the errors in the actions.

Motivated by Theorem 4, maximizing the global radius amounts to minimizing the occurrence of per-step errors in policy predictions. From Equation 16 in Theorem 4, it follows that maximizing the local certified radius hinges on increasing the gap between the top-1 and the runner-up Q-values under randomized observations. We can thus define a robustness loss such that the per-step action error is minimized:

$$\begin{aligned} \ell_{ro}(\pi, \tilde{\pi}; s, \epsilon) &= \lambda \cdot \mathbb{1}_{\{Q_\pi(s+\epsilon, a^{(1)}) \geq Q_{\tilde{\pi}}(s+\epsilon, a^{(2)})\}} \\ &\quad \max\{0, \eta - [Q_\pi(s+\epsilon, a^{(1)}) - Q_{\tilde{\pi}}(s+\epsilon, a^{(2)})]\}, \\ a^{(1)} &= \arg \max_a \pi(s+\epsilon)_a, \\ a^{(2)} &= \arg \max_{a: a \neq a^{(1)}} \pi(s+\epsilon)_a. \end{aligned} \quad (17)$$

Herein,  $\eta$  and  $\lambda$  represent the hinge loss offset and the robustness loss coefficient, respectively.  $s$  denotes an observation, which may be deterministic or randomized.  $a^{(1)}$  and  $a^{(2)}$  are the predicted top-1 and runner-up actions in the action space  $\mathbb{A}$ . Note that the predicted action may differ from the optimal  $a^{(1)}$  and  $a^{(2)}$  required by Theorem 4. To mitigate this,  $\tilde{\pi}$  is introduced as a *reference network* to check whether  $a^{(1)}$  indeed yields a higher reward than  $a^{(2)}$ . The details of  $\tilde{\pi}$  will be discussed in the following section. This robustness loss regularizes policy training to maximize the gap between the Q-value induced by the top-1 action and that obtained by the runner-up action.

**Remark.** Our certified radii differ from existing approaches in several aspects. First, they are functions of optimizable variables instead of a fixed  $\tau$  determined via grid search. Second, by framing the total robustness cost as the cumulative return loss from per-step action errors, we convert the global certified radius into a local certified radius linked to the Q-gap. Maximizing the Q-gap at each step reduces action errors, thereby enhancing the global certified radius.

### 4.3 Policy Imitation for CAMP Training

Given the objectives of improving both utility and robustness, the training objective of CAMP can be defined as minimizing the sum of the utility and robustness losses. The remaining task is to determine the reference network  $\tilde{\pi}$ . We found that attaching the robustness loss to the TD loss during training is unsuitable, as it leads to overestimation of Q-values.

Overestimating Q-values is a well-known issue that leads to sub-optimal convergence [63]. Moreover, it is infeasible to constrain the maximal Q-value of the Q-network since the oracle Q-value is unknown. This dilemma motivates us to first approximate the oracle Q-value and use the approximated value as a reference to calibrate the Q-network.

To this end, we propose *policy imitation* to stabilize the training process. Specifically, we model the normal Q-function with a reference Q-network whose parameters are denoted as  $\tilde{\pi}$ .  $\tilde{\pi}$  is trained using the utility loss  $\ell_{ut}(\tilde{\pi}; s, \epsilon, a, s', \epsilon')$ . The primary Q-network then mimics the behavior of this reference network to improve its utility while the robustness loss is applied for robustness enhancement. In this way, the reference network helps the primary Q-network predict more accurate Q-values. The imitation loss is defined as the cross-entropy loss between  $\text{Softmax}(\pi(s + \epsilon))$  and  $\text{Softmax}(\tilde{\pi}(s + \epsilon))$ :

$$\ell_{im}(\pi, \tilde{\pi}; s, \epsilon) = - \sum_{i=1}^{|\mathbb{A}|} \text{Softmax}(\tilde{\pi}(s + \epsilon))_i \log \text{Softmax}(\pi(s + \epsilon))_i. \quad (18)$$

Therefore,  $\ell_{im}(\pi, \tilde{\pi}; s, \epsilon)$  measures the difference between the actions predicted by  $\pi$  and  $\tilde{\pi}$ , given the same observation input  $s + \epsilon$ .

The robustness loss and the imitation loss take the same randomized observation  $s + \epsilon$  as used in the utility loss for computation. The final loss function of CAMP is as follows:

$$\ell_{CAMP}(\pi, \tilde{\pi}; s, \epsilon, s', \epsilon') = \ell_{ut}(\tilde{\pi}; s, \epsilon, s', \epsilon') + \ell_{ro}(\pi, \tilde{\pi}; s, \epsilon) + \ell_{im}(\pi, \tilde{\pi}; s, \epsilon). \quad (19)$$

Let  $\mathcal{Z}$  denote a replay buffer or a dataset of trajectories. The overall training objective is then defined as follows:

$$\min_{\pi, \tilde{\pi}} \mathbb{E}_{(s, \epsilon, s', \epsilon') \sim \mathcal{Z}} \ell_{CAMP}(\pi, \tilde{\pi}; s, \epsilon, s', \epsilon'). \quad (20)$$

In our implementation, we found that using two separate replay buffers for  $\pi$  and  $\tilde{\pi}$  can enhance the agent's performance and stability. Consequently,  $\pi$  and  $\tilde{\pi}$  operate alternately in the same environment, storing the collected trajectories into two distinct replay buffers,  $\mathcal{Z}$  and  $\tilde{\mathcal{Z}}$ . The training algorithm samples trajectories from  $\tilde{\mathcal{Z}}$  to minimize the robustness loss and the imitation loss, while sampling from  $\mathcal{Z}$  to minimize the utility loss.

To summarize the practical training steps of CAMP, the complete training procedure is presented in Algorithm 1. Note that  $d_t$ ,  $d_j$  and  $\tilde{d}_j$  are binary variables representing whether the current states (*i.e.*,  $s_t$ ,  $s_j$ , and  $\tilde{s}_j$ ) are terminal states. If yes, the Q-function should show that the agent gets no additional rewards after the current state. Moreover, only the derivatives of  $\ell_{ut}$  with respect to the reference network parameters are calculated in line 25. Conversely, since  $\ell_{ut}$  is independent of the primary Q-network, only  $\ell_{ro}$  and  $\ell_{im}$  induce gradients for optimizing the primary network in line 26. We will validate the effectiveness of CAMP through experiments in the following section.

---

### Algorithm 1: CAMP Training

---

**Input:** Replay buffers  $\mathcal{Z}$  and  $\tilde{\mathcal{Z}}$ , environment  $\mathcal{E}$  with state transition function  $\mathcal{T}$  and reward function  $\mathcal{R}$ , discount factor  $\gamma$ , primary network  $\pi$ , reference network  $\tilde{\pi}$ , noise scale  $\sigma$ , polyak rate  $k$ , burn in step  $t_b$ , batch size  $N$ , learning rate  $\alpha$ , target update frequency  $\Delta$ .

**Output:** Trained  $\pi$

```

1 Initialize  $s_0$ 
2 Initialize  $\pi, \tilde{\pi}$ 
3 Initialize reference target network  $\tilde{\pi}' \leftarrow \tilde{\pi}$ 
4 for  $t \in \{0, 1, 2, \dots, T-1\}$  do
5    $\epsilon_t, \epsilon_{t+1} \sim \mathcal{N}(0, \sigma^2 I)$ 
6    $\tilde{s}_t \leftarrow s_t + \epsilon_t$ 
7   if  $t \leq t_b$  then
8      $a_t \leftarrow \text{RANDOMSAMPLE}(\mathbb{A})$ 
9      $s_{t+1}, d_t, r_{t+1} \leftarrow \mathcal{T}(s_t, a_t), \mathcal{R}(s_t, a_t)$ 
10    if  $t \% 2 = 0$  then
11       $\mathcal{Z} \leftarrow (s_t, \epsilon_t, s_{t+1}, \epsilon_{t+1}, d_t)$ 
12    else
13       $\tilde{\mathcal{Z}} \leftarrow (s_t, \epsilon_t, s_{t+1}, \epsilon_{t+1}, d_t)$ 
14  else
15    if  $t \% 2 = 0$  then
16       $a_t \leftarrow \arg \max_a \pi(s_t)_a$ 
17       $s_{t+1}, d_t, r_{t+1} \leftarrow \mathcal{T}(s_t, a_t), \mathcal{R}(s_t, a_t)$ 
18       $\mathcal{Z} \leftarrow (s_t, \epsilon_t, s_{t+1}, \epsilon_{t+1}, d_t)$ 
19    else
20       $a_t \leftarrow \arg \max_a \tilde{\pi}(s_t)_a$ 
21       $s_{t+1}, d_t, r_{t+1} \leftarrow \mathcal{T}(s_t, a_t), \mathcal{R}(s_t, a_t)$ 
22       $\tilde{\mathcal{Z}} \leftarrow (s_t, \epsilon_t, s_{t+1}, \epsilon_{t+1}, d_t)$ 
23     $\{(s_j, \epsilon_j, s'_j, \epsilon'_j, d_j)\}_{j=1}^N \leftarrow \text{BATCHSAMPLE}(\mathcal{Z}, N)$ 
24     $\{(\tilde{s}_j, \tilde{\epsilon}_j, \tilde{s}'_j, \tilde{\epsilon}'_j, \tilde{d}_j)\}_{j=1}^N \leftarrow \text{BATCHSAMPLE}(\tilde{\mathcal{Z}}, N)$ 
25     $\tilde{\pi} \leftarrow \tilde{\pi} - \alpha(1 - \tilde{d}_j) \frac{d \sum_{j=1}^N \ell_{CAMP}(\pi, \tilde{\pi}; \tilde{s}_j, \tilde{\epsilon}_j, \tilde{s}'_j, \tilde{\epsilon}'_j)}{d \tilde{\pi}}$ 
26     $\pi \leftarrow \pi - \alpha(1 - d_j) \frac{d \sum_{j=1}^N \ell_{CAMP}(\pi, \tilde{\pi}; s_j, \epsilon_j, s'_j, \epsilon'_j)}{d \pi}$ 
27    if  $t \% \Delta == 0$  then
28       $\tilde{\pi}' \leftarrow k \tilde{\pi} + (1 - k) \tilde{\pi}'$ 

```

**Output:**  $\pi$

---

## 5 Experiments

We evaluate the performance of CAMP in this section. Our experiments aim to answer the following research questions:

- Can CAMP improve the certified expected returns under the same perturbation budget?
- Is CAMP a broadly applicable enhancement for various environments and Q-networks?
- How robust is CAMP towards changes in its hyper-parameters?

To answer these questions, we systematically compare CAMP to baselines in different environments and conduct an ablation study on the hyper-parameters. We will first introduce the environments, DRL algorithms, Q-networks, and certification methods used in our experiments.

**Environments.** We make evaluations and comparisons in five representative environments, namely *Cartpole*, *Highway*, *Bank Heist*, *Pong*, and *Freeway*. Cartpole simulates a classic



problem in control dynamics. The observed states in Cartpole are four kinematic signals representing the cart position, cart velocity, pole angle, and pole angular velocity. The action space contains two discrete actions (*i.e.*, push the cart to the left or right). Highway simulates an autonomous driving environment in which the observations are based on the kinematics of nearby cars and the actions are four control inputs. Freeway, Pong and Bank Heist are Atari games that examine the performance of certification algorithms towards high-dimensional observations. The observed states in both environments are frames of  $84 \times 84$  greyscale images with pixel values in  $[0, 255]$ . The available actions in Freeway, Pong, and Bank Heist consist of 3, 6, and 18 discrete options, respectively. We adopt Pong with only one round (*i.e.*, Pong1r) in our experiments.

**DRL algorithms.** We evaluated our method based on DQN in various environments. DQN is employed as the DRL algorithm for evaluation. For CartPole and Highway, the architecture of the Q-network is a multi-layer perceptron (MLP) network. On the other hand, Nature CNN is employed as the Q-network for Freeway, Pong, and Bank Heist [38]. The detailed architectures are also attached in Appendix B.3. These settings follow the convention in the evaluation of DQN, which endorses fair comparisons with previous works.

**Certification algorithms and baselines.** We employ Policy-Smoothing (PS) [23] as the certification algorithm. PS leverages the generalized Neyman-Pearson lemma towards the structure deterministic adversary and generates tight robustness certificates. We apply the exact CDF smoothing as introduced in Section 2.2 for agents obtaining continuous rewards in Highway, Freeway, and Bank Heist. Following the PS paper, for agents obtaining 0/1 reward at each time step in Cartpole and Pong1r, we employ per-step point estimates of the CDF function based on the Clopper-Pearson method [5] and take the sum as the CDF. This method is equivalent to CDF smoothing and is found to achieve slightly better results in practice. We set two baselines by certifying using policies trained by Gaussian Augmentation (Gaussian) [23, 60] and NoisyNet [44]. Gaussian injects noises sampled from the same Gaussian distribution used for certification into the observations during training. On the other hand, NoisyNet randomizes the weights of the linear layers in the Q-network to aid efficient exploration during training and improve return obtained during tests. We apply noise with the same scale to all the methods during training for comparison.

**Hyper-parameters and detailed settings.** The discount factor in all training experiments is set to  $\gamma = 0.99$  except for Highway, where  $\gamma = 0.8$ . The agents randomly sample actions according to a greedy probability which decreases linearly from 1 to 0 within the first 16% training steps. In Highway, the probability decreases linearly from 1 to 0.05 over the first 10% of the training steps. The target networks are updated softly using a Polyak rate of 1. In CAMP, we use adaptive  $\eta$  values by setting  $\eta = \max_{(s,a) \sim \mathcal{Z}_t} Q_\pi(s,a) - \min_{(s,a) \sim \mathcal{Z}_t} Q_\pi(s,a)$ ,

where  $\mathcal{Z}_t$  represents the trajectory batch at the  $t$ -th step. We also conduct an ablation study in Section 5.2 to analyze the impact of training with different  $\lambda$  values on the certification performance. In the certification, the return is computed with  $\gamma = 1$ . Each certification run includes return values from 10000 game rounds in which the observation is randomized by Gaussian noise.

- **Cartpole:** We train agents in the Gymnasium implementation of *CartPole-v0* with single-frame observations (Cartpole-1) and five-frame observations (Cartpole-5), respectively. In both cases, we use a batch size of 1024, a learning rate of  $5 \times 10^{-5}$ , and the training spans 500k steps. The training stops when the return reaches 200 (*i.e.*, the highest possible reward). The agent starts training after 10000 burn-in steps, and the target network is updated every 10 steps.
- **Highway:** We select *highway-fast-v0* as the environment. We use a batch size of 1024, a learning rate of  $5 \times 10^{-5}$ , and train for 1000k steps. The training starts after 1000 steps, and the target network is updated every 10 steps.
- **Atari:** We use Gymnasium implementations of *FreewayNoFrameskip-v0* in hard mode, *PongNoFrameskip-v0*, and *BankHeistNoFrameskip-v4* as the implementations. The training takes 10 million steps with a learning rate of  $1 \times 10^{-4}$ . We observed that applying policy imitation at a later training stage can enhance training stability. Therefore, in our experiments, we first train the reference network until convergence. Subsequently, we duplicate the reference network as the primary network and apply policy imitation with CAMP to update the primary network while keeping the reference network fixed. The reference network is trained under the same setting as the baselines while updating the primary policy takes an extra 1 million steps.

Detailed hyper-parameter settings are given in Appendix B.1.

## 5.1 Evaluation

**Comparison of certification results.** We benchmark the certified reward of our method with that from baselines in this section. We unify the hyper-parameters in the certification and apply them to agents trained by Gaussian, NoisyNet, and CAMP, respectively. For clarity, in Cartpole-1 and Cartpole-5, we compare the certified returns when  $\sigma \in \{0.2, 0.6, 1.0\}$  and plot them in Figure 2. Meanwhile, the full Cartpole certification results based on  $\sigma \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$  are provided in Figure 8 of Appendix B.2. We compare all three methods in these two environments.

Based on the results, the certified expected returns from CAMP show universal increases across all combinations of certified radii and smoothing noise levels compared to the baselines. In both Cartpole-1 and Cartpole-5, the increase in certified expected returns with CAMP becomes more pronounced when certifying at an  $\ell_2$  radius between 0.2 and

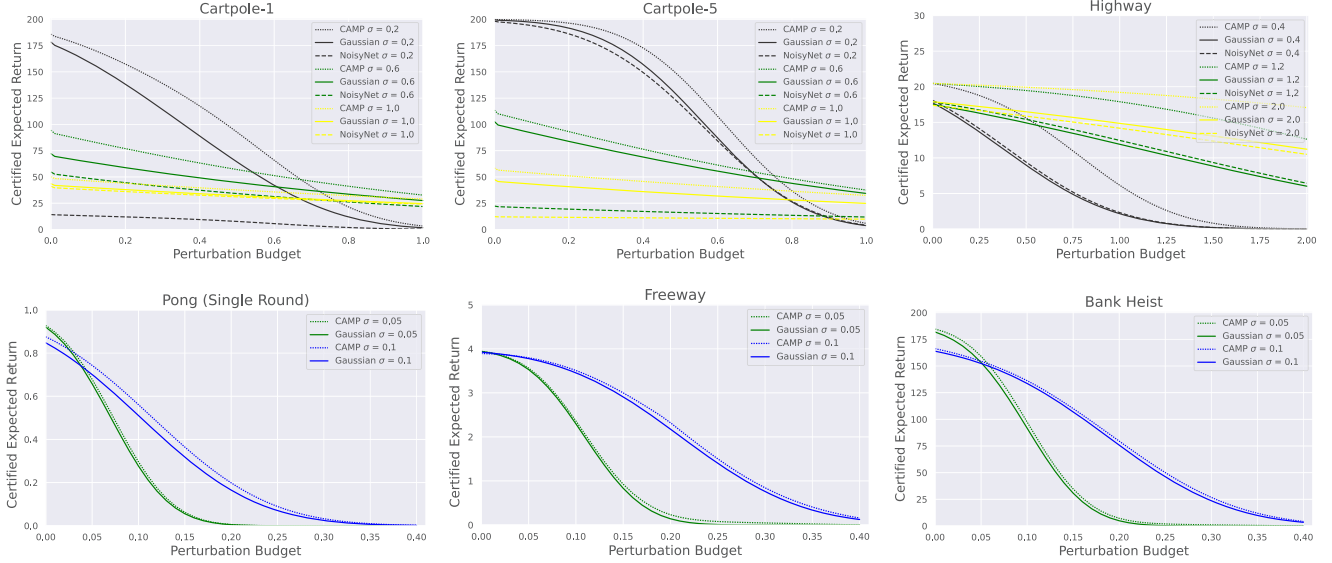


Figure 2: Certification results on CartPole, Highway, Pong, Freeway, and Bank Heist. The perturbation budgets for Atari games (Freeway, Pong, and Bank Heist) are normalized by dividing by 255.

0.8. According to Figure 2, in Cartpole-1, the utility gain from CAMP is slightly more obvious than that in Cartpole-5. However, the most significant improvement is observed in Cartpole-5 at  $\sigma = 0.4$ , where CAMP doubles the certified return against adversaries with budgets between 0.2 and 1.0. These results indicate that CAMP can effectively enhance certified returns at fixed certified radii in environments with low-dimensional observations. NoisyNet performs worse than both Gaussian and CAMP, suggesting that NoisyNet may be ineffective in handling observations with significant noise. Notably, when  $\sigma = 0.2$  in Cartpole-1, NoisyNet fails in certification, likely due to underfitting caused by its randomized parameters, which complicates learning the Q-function in conditions with less informative observations.

Additionally, the results for the Highway environment with  $\sigma \in \{0.4, 1.2, 2.0\}$  are shown in Figure 2. It can be observed that CAMP significantly outperforms the baselines across all smoothing noise levels. These results suggest that CAMP agents are relatively robust in the Highway environment and can tolerate more adversarial perturbations compared to agents trained by the baseline methods. The full comparison with  $\sigma \in \{0.4, 0.8, 1.2, 1.6, 2.0\}$  is included in Appendix B.2.

On the other hand, we certify agents in Freeway, Pong1r, and Bank Heist with  $\sigma \in \{12.75, 25.5\}$  to evaluate the effectiveness of CAMP in high-dimensional observation spaces. Improving certification performance in these high-dimensional environments is considered a more challenging task. We include Gaussian and CAMP in the comparison, as NoisyNet consistently underperformed Gaussian in our experiments when training in noisy environments. Similarly, the certification results are in Figure 2. In the plots, we normalize the perturbation budget and  $\sigma$  values by dividing by 255. It is

evident that CAMP consistently certifies better expected returns than Gaussian. While the improvements are more subtle compared to other environments, they are particularly noticeable when the attack budget ranges from  $[11.25, 63.75]$  in Pong,  $[38.25, 102]$  in Freeway, and  $[0, 76.5]$  in Bank Heist.

Given these certification results on expected returns, we proceed to examine the empirical robustness of return values in each game round against adversarial perturbations.

**Empirical robustness verification.** In addition to comparing certification results, we also explore the empirical robustness of the trained DRL agent in each game episode. Recall that certification requires running the agent in the game numerous times to compute a lower bound on the expected return. Beyond this, we are also interested in the robustness of the agent in single game runs against adversaries. To this end, we extend the adversarial attack methods from previous literature [23, 36] to evaluate CAMP.

Given a trained policy  $\pi$ , the attack perturbs the observed state  $s$  by a perturbation  $\delta$  to generate a misleading action  $a' = \arg \max_a \pi(s + \delta)_a$ . Compared to the original action  $a^* = \arg \max_a \pi(s)_a$  returned based on the unperturbed observation,  $a'$  is a minimizer of the original Q-value  $Q_\pi(s, a')$ . Specifically, we use Projected Gradient Descent (PGD) [37] and AutoAttack with AutoPGD (APGD) [6] to generate perturbations constrained by  $\ell_2$ -norm budgets. In the attacks, we minimize the Cross-Entropy loss between  $\pi(s + \delta)$  and (the one-hot vector of) a target action  $a'$  from the action space  $\mathbb{A}$  and iteratively search for the  $a'$  that produces the lowest  $Q_\pi(s, a')$  value across the action space. The total  $\ell_2$  budget is fixed, and any remaining budget from the current time step rolls over to the next. The attack at each time step terminates if  $\arg \max_a \pi(s + \delta)_a$  equals  $a'$  or the perturbation

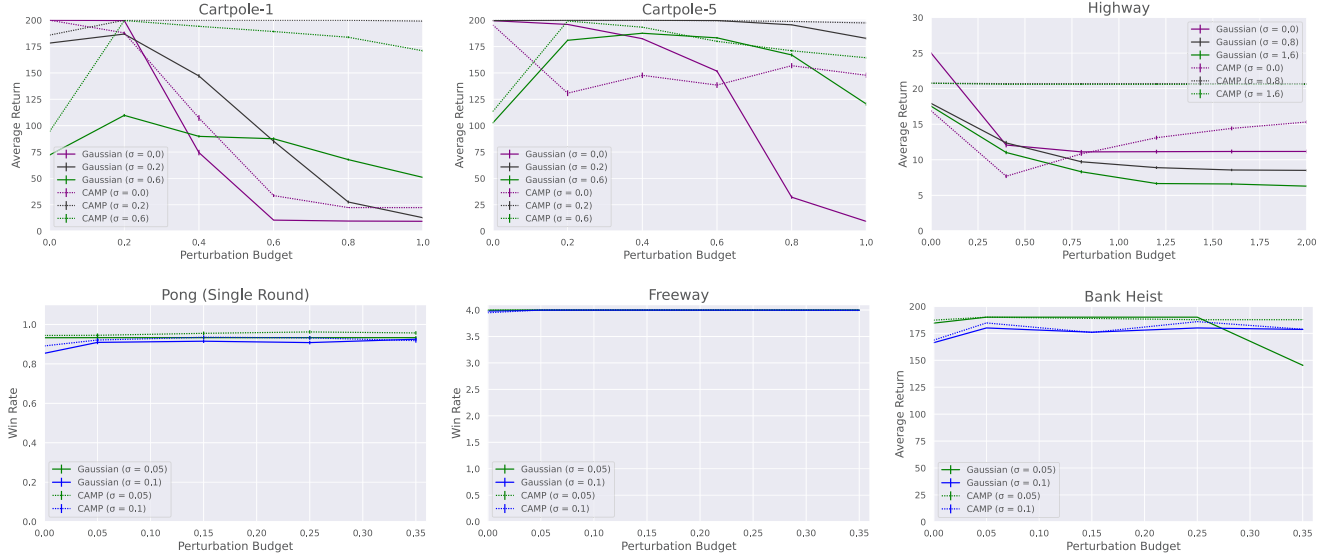


Figure 3: Empirical robustness of agents against PGD in CartPole, Highway, Pong, Freeway, and Bank Heist. We use the same attack in PS [23] to evaluate the robustness of the agent in individual runs. The perturbation budgets for Freeway, Pong, and Bank Heist are normalized by dividing by 255. In Pong, since agents either win or lose in each run, the expected return corresponds to the win rate.

budget is exhausted. The detailed algorithms are presented in Appendix C.

We measure empirical robustness by the average return from 1000 independent episode plays. The average returns of different agents across various environments under different perturbations are illustrated in Figures 3-4, with error bars indicating variability. Notably, agents under attack demonstrate higher performance than the certified results shown in Figures 2 and 8. This discrepancy arises because 1) certified expected returns represent worst-case scenarios and serve as lower bounds for the average returns in Figures 3-4, and 2) empirical attacks consume the perturbation budget from the initial time step rather than strategically allocating perturbations to steps causing the most significant reward loss.

CAMP outperforms the baseline in most of the evaluated scenarios with non-zero Gaussian noise augmentations. In both Figure 3 and Figure 4, we first assess the Cartpole-1/Cartpole-5 agents trained with noisy observations, where  $\sigma \in \{0, 0.2, 0.6\}$ , and subjected to perturbations with  $\ell_2$  budget caps  $\tau$  from  $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ . A Gaussian agent with  $\sigma = 0$  serves as a baseline representing a completely undefended agent, while Gaussian agents with  $\sigma \in \{0.2, 0.6\}$  are trained using the Gaussian baseline defense under moderate to high noise levels. In comparison, CAMP agents with  $\sigma \in \{0.2, 0.6\}$  are evaluated against their corresponding Gaussian baselines to highlight improvements in robustness. In the undefended scenario, agents trained with CAMP still exhibit greater robustness when facing large perturbations.

Similarly, we apply attacks to Gaussian and CAMP agents in Highway, selecting  $\sigma \in \{0, 0.8, 1.6\}$  and attack budgets

Table 1: Return of Agents Trained in Noise-Free Environments

Game	Method		
	CAMP	Gaussian	NoisyNet
Cartpole-1	200.00	199.94	200.00
Cartpole-5	195.81	199.50	199.68
Pong1r	0.91	0.90	-
Freeway	4.00	4.00	-

$\tau \in \{0.4, 0.8, 1.2, 1.6, 2.0\}$ . Compared to Gaussian agents, CAMP agents are highly robust against both PGD and APGD attacks. According to the results, all Gaussian agents experience significant performance degradation under attacks with  $\tau \geq 0.4$ , whereas CAMP agents trained with  $\sigma \in \{0.2, 0.6\}$  remain unaffected. Even when  $\sigma = 0$ , the CAMP agent exhibits superior robustness under attack budgets of  $\tau \geq 0.8$  against PGD and  $\tau \geq 1.2$  against APGD.

In Atari games, we set  $\sigma \in \{12.75, 25.5\}$  and the perturbation budgets are chosen from  $\{12.75, 38.25, 63.75, 89.25\}$ . Both  $\sigma$  and perturbation budget values are normalized to the range  $[0, 1]$  in the plot for Atari games. Agents in Atari games are generally more robust against both attacks. Augmenting observations with noise enhances agent robustness in both the Gaussian and CAMP frameworks. However, the average return of Gaussian agents is more prone to observation noise and declines rapidly with increasing attack budgets in Bank Heist. In contrast, CAMP effectively mitigates this impact and maintains a higher average return across diverse attack scenarios. These results indicate that CAMP not only advances reward certification but also improves the robustness of Gaussian-augmented DRL agents against empirical attacks in individual episodes.

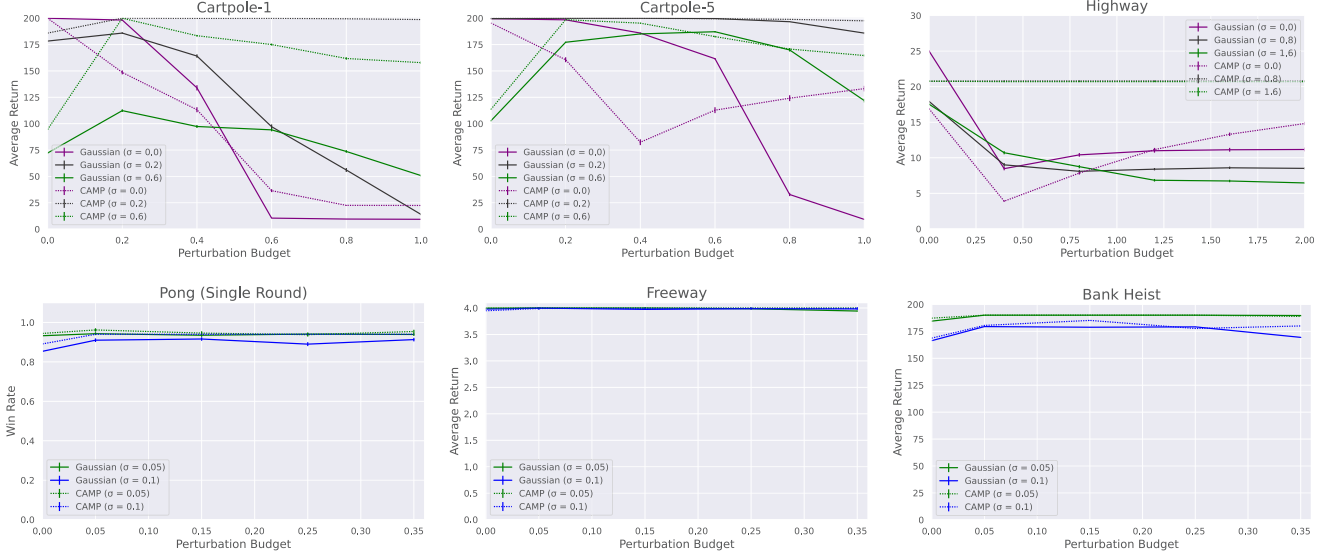


Figure 4: Empirical robustness of agents against APGD in CartPole, Highway, Pong, Freeway, and Bank Heist. The average return values are evaluated under the same settings as PGD. APGD preserves the perturbation budget at each step, allowing it to perturb observations across more steps, which can result in more significant performance degradation for the agents.

Table 2: Minimal Q-gap under Varying  $\lambda$  and  $\sigma$  Values

Game	Method	$\sigma$					
		0.0	0.2	0.4	0.6	0.8	1.0
Cartpole-1	Gaussian	0	0	0	0	0	0
	NoisyNet	0	0	0	0	0	0
	CAMP( $\lambda = 0.5$ )	$1.49 \times 10^{-8}$	$6.37 \times 10^{-7}$	$1.25 \times 10^{-5}$	$4.55 \times 10^{-6}$	$2.68 \times 10^{-5}$	$1.30 \times 10^{-7}$
	CAMP( $\lambda = 1$ )	$2.91 \times 10^{-7}$	$2.89 \times 10^{-6}$	$9.82 \times 10^{-6}$	$9.57 \times 10^{-5}$	$2.10 \times 10^{-5}$	$4.82 \times 10^{-6}$
	CAMP( $\lambda = 4$ )	$1.86 \times 10^{-7}$	$2.98 \times 10^{-8}$	$4.10 \times 10^{-7}$	$2.52 \times 10^{-5}$	$2.46 \times 10^{-5}$	$5.58 \times 10^{-9}$
	CAMP( $\lambda = 16$ )	0	$2.24 \times 10^{-8}$	$1.86 \times 10^{-9}$	0	$9.31 \times 10^{-9}$	$3.73 \times 10^{-9}$
Cartpole-5	Gaussian	0	0	0	0	0	0
	NoisyNet	0	0	0	0	0	0
	CAMP( $\lambda = 0.5$ )	$6.71 \times 10^{-8}$	$2.48 \times 10^{-6}$	$3.24 \times 10^{-6}$	$9.24 \times 10^{-7}$	$2.18 \times 10^{-6}$	$1.30 \times 10^{-6}$
	CAMP( $\lambda = 1$ )	$3.35 \times 10^{-8}$	$1.19 \times 10^{-7}$	$1.11 \times 10^{-5}$	$3.43 \times 10^{-7}$	$6.12 \times 10^{-5}$	$3.99 \times 10^{-5}$
	CAMP( $\lambda = 4$ )	$2.94 \times 10^{-7}$	$1.86 \times 10^{-9}$	$4.84 \times 10^{-8}$	$3.51 \times 10^{-5}$	$2.02 \times 10^{-4}$	$6.33 \times 10^{-8}$
	CAMP( $\lambda = 16$ )	$7.45 \times 10^{-9}$	$5.59 \times 10^{-9}$	0	$5.59 \times 10^{-9}$	$1.10 \times 10^{-6}$	$4.47 \times 10^{-8}$

Table 3: Training Time for Different Methods

Game	Time (GPU Hour)		
	CAMP	Gaussian	NoisyNet
Cartpole-1	0.98	0.47	0.62
Cartpole-5	0.93	0.42	0.53
Highway	13.78	7.00	6.98
Pong1r	27.91	12.73	-
Freeway	29.75	14.60	-
Bank Heist	20.11	19.63	-

**Impact on training in normal environments.** We also evaluated the performance of CAMP when training DRL agents in environments without observation noise. We test the trained agents and record their test returns in Table 1. Each return in the table is averaged from 100 independent game runs. It can be observed that CAMP does not degrade the training performance in noise-free environments, supporting the versatility of CAMP.

## 5.2 Ablation Studies

In this section, we investigate the impact of  $\lambda$  on the trade-off between the certified expected return and certified radius, the

effect of CAMP on the policy’s Q-gap, and the convergence of the training.

**The effect of  $\lambda$ .** First, we are interested in investigating how the coefficient  $\lambda$  impacts the training outcome. Herein, we train DQN agents on Cartpole-1 and Cartpole-5 using  $\lambda \in \{0.5, 1, 2, 4, 8, 16\}$  in the training. The certification results based on the trained agents are illustrated in Figure 5 and Figure 6. It can be observed that, generally, the certified performance stays constant over varying  $\lambda$  values. However, the performance drops in a few cases. Especially, small  $\lambda$  for large noise scale or large  $\lambda$  for small noise result in the decreases.

**Convergence of training .** We visualize the mean episode reward during the validation for CAMP and Gaussian, respectively. The environment noise levels in our visualization are based on  $\sigma \in \{0.0, 0.4, 1.0\}$  for Cartpole-1 and Cartpole-5, such that the trends of training in both noise-free and rather noisy environments can be observed. The validation rewards are averaged from playing 10 runs of games after every 2000 training steps, and the results are plotted in Figure 7. According to the figure, CAMP reaches the maximal return at the same pace with or slightly slower than Gaussian when  $\sigma = 0$ . However, when  $\sigma$  increases, CAMP can converge to higher mean validation return values in earlier training stages.

On Atari games, the convergence of the reference and primary policies is visualized in separate sub-figures in Figure 7, based on Freeway and Pong1r. The agent with the reference network undergoes validation after every 100k training steps, while the primary agent is validated every 1000 training steps. Each validation reward is averaged from 100 episodes. The primary policies experience an initial drop in validation return



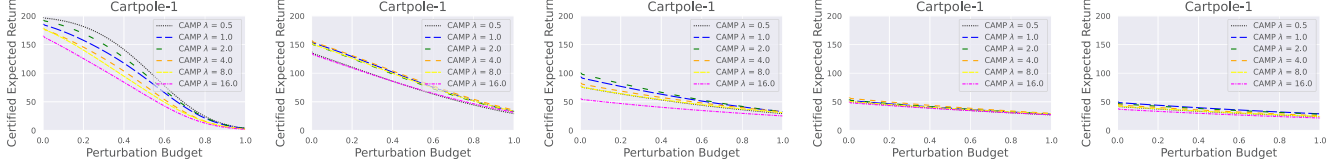


Figure 5: Ablation on  $\lambda$  values in Cartpole-1. The certified expected returns obtained from various  $\lambda$  values are shown in the figures. Each figure presents results based on a fixed smoothing noise scale applied to the observed states. From left to right, the smoothing noise scales are 0.2, 0.4, 0.6, 0.8, and 1.0, respectively.

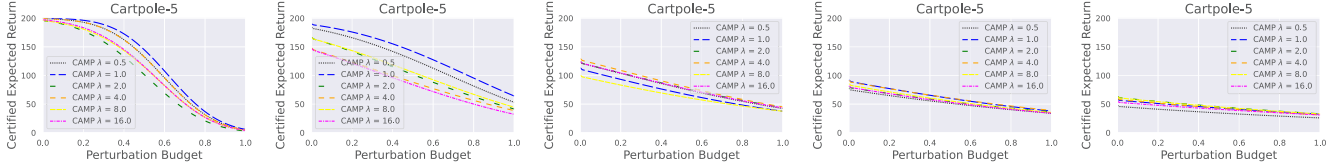


Figure 6: Ablation on  $\lambda$  values in Cartpole-5. The certified expected returns obtained from various  $\lambda$  values are shown in the figures. Each figure presents results based on a fixed smoothing noise scale applied to the observed states. From left to right, the smoothing noise scales are 0.2, 0.4, 0.6, 0.8, and 1.0, respectively.

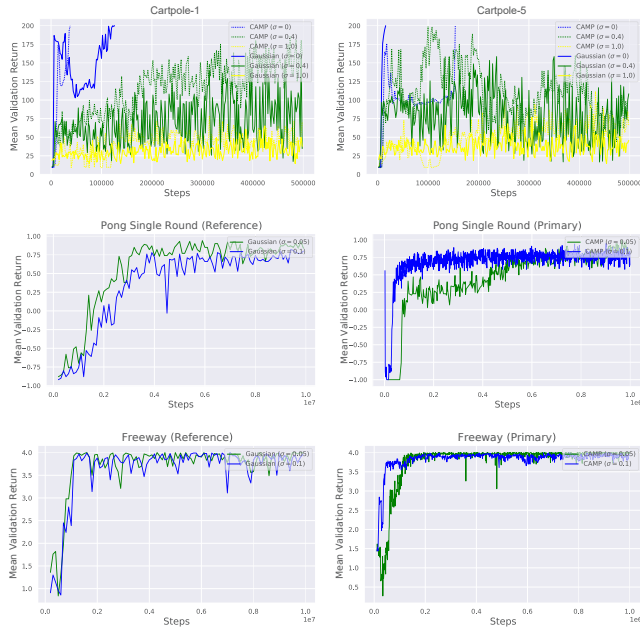


Figure 7: Validation returns in different environments.

due to the application of the new loss function. However, both policies quickly recover, with validation rewards stabilizing before 200k training steps and converging by 1 million steps.

**The gap between top-1 and runner-up Q-values.** Finally, we investigate the gap between the top-1 action score and the runner-up action score given state observation  $s$  under different  $\sigma$  settings. According to Theorem 4, larger Q-gaps lead to greater certified local radii and thus make it harder for the adversary at the current step to manipulate the action through observation perturbations. Particularly, the minimum of the Q-gap is the most crucial statistic signifying the robustness

of the agent in worst cases. Therefore, we would like to examine whether CAMP can effectively widen this minimum of the Q-gap. In our experiments, we select CAMP agents trained with  $\sigma \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$  and  $\lambda \in \{0.5, 1, 4, 16\}$  to compare with agents trained by Gaussian and NoisyNet. Each trained agent plays the same 10000 runs of games and records the minimum of its Q-gap values, as shown in Table 2. It can be observed that CAMP effectively upscales the minimal Q-gap and thus makes the agent more robust to noise in the environments.

### 5.3 Computational Cost and Scalability

We document the training costs of CAMP in Table 3, where all experiments were conducted using Nvidia H100 GPUs. We measured the GPU hours required for training CAMP and compared them to the baseline methods. The time costs for Cartpole-1 and Cartpole-5 are from the cases where  $\sigma = 1.0$  and the costs for training in Atari games are based on  $\sigma = 0.1$ . It is important to note that training typically requires more time to converge under higher levels of observation noise, so the numbers reflect the longest training times for the various methods in each environment. Note that the time duration includes the computational cost of validation steps.

Although CAMP increases the training duration of DRL agents, the additional time remains within acceptable limits. First, data parallelism with multiple GPUs can effectively reduce the actual training time. In particular, when training in environments with high-dimensional observations, CAMP distills its Q-network from a pretrained reference network with a batch size of 32. This process can be further accelerated by using larger batch sizes during distillation. Furthermore, CAMP remains scalable to high-dimensional action spaces. Specifically, we use Cross-Entropy for the imitation loss, as it scales

efficiently to large category sizes. Additionally, the robustness loss relies only on the top-2 Q values, ensuring mathematical scalability to high-dimensional action spaces.

**Main takeaways.** CAMP effectively enhances both the certified expected return and empirical robustness of DQN agents. Significant performance gains are observed in classic control and autonomous driving environments (e.g., Cartpole and Highway), while Atari agents, despite showing subtler improvements, are inherently more robust against empirical attacks. Further improvements in Atari games might be achieved by training the primary Q-network from reference network checkpoints instead of distilling from a fully trained reference Q-network. Additionally, CAMP is resilient to hyper-parameter variations and scales effectively to large action spaces.

## 6 Related Work

**Adversarial attack and robust DRL.** DRL has been demonstrated to be vulnerable to adversarial attacks. Early attacks adapted adversarial strategies from neural network classifiers to perturb the observations of DRL agents [4, 16, 22]. Later, attacks were developed to degrade agent performance through interactive adversarial games [12, 41, 43]. Other attacks have exploited vulnerabilities across different steps [54] and used policy-independent perturbations [21]. Beyond targeting a single DRL agent, adversaries have also sought to compromise multi-agent DRL systems [33]. The robustness of DRL has garnered substantial attention in recent years. Building on early  $H_\infty$  robust control theory for worst-case performance under deterministic dynamics [3], methods have emerged that apply robust control to Markov Decision Processes (MDPs) by modeling uncertainty in transition matrices [15, 40, 45]. Additionally, policy smoothing and observation smoothing have been advocated to improve the robustness of DRL [48, 51, 63]. More recently, some approaches have addressed adversarial robustness in offline DRL through action randomization [20, 43], and several works have incorporated adversarial training as a defense mechanism [35, 67, 68]. Furthermore, there are defenses against adversarial policies in multi-agent DRL [13, 34]. Importantly, significant efforts have been made towards providing theoretical robustness guarantees for DRL [29, 64, 69]. However, these approaches often rely on specific assumptions and are not easily scalable to high-dimensional DRL problems.

**Randomized smoothing and certifiably robust DRL.** RS has emerged as a practical tool for providing certified robustness, building on previous works that relied on differential privacy and Rényi divergence [5, 26, 28]. Numerous efforts have been made to certify  $\ell_p$  robustness radii for classifiers, using approaches such as the Neyman-Pearson lemma and beyond [5, 7, 8, 11, 14, 30, 47, 52, 66]. RS has also been ex-

tended to certify the learnability of unlearnable data [58]. Additionally, the limitations of RS against  $\ell_p$  ( $p > 2$ ) adversaries in high-dimensional input spaces have been identified and are being addressed [25, 31, 49, 61]. In the context of reinforcement learning, three closely related works certify a lower bound on the returns of DRL agents through randomized policies [23, 39, 60]. Several studies have also explored the certified robustness of multi-agent DRL [55]. A recent work applies denoising smoothing to improve the utility of provably robust DRL agents [53]. However, the development of methods for obtaining policies specifically tailored for RS has been overlooked in the current literature.

## 7 Conclusion

We propose CAMP, a method designed to train DRL policies with enhanced certified utility and robustness for policy-smoothing-based robustness certification. This approach is grounded in the analysis of certified lower bounds of expected returns obtained by smoothed policies. We reformulate the certification problem by decomposing it into two sub-problems which are maximizing expected return and maximizing certified radius. While maximizing the expected return is straightforward, maximizing the certified radius presents challenges due to the lack of a differentiable, closed-form expression for the radius. We address this challenge by converting the certified radius into a soft radius and further transforming it into a per-step radius, which characterizes the maximum perturbation that does not degrade the current optimal expected return. Through experiments, we have verified the effectiveness of CAMP in improving certified expected returns at fixed certified radii in environments with both simple and high-dimensional observations. Additionally, CAMP agents demonstrate superior robustness against empirical adversarial perturbations in each game run. Nevertheless, there are limitations to be addressed. CAMP currently supports only discrete action spaces. Additionally, it selects the top-1 and runner-up Q-values based on a presumed optimal policy, which may not be fully captured by the reference network. Moreover, CAMP exhibits less pronounced improvements in environments with high-dimensional visual observations compared to control and autonomous driving scenarios. To address these limitations, future work will extend CAMP to continuous action spaces, refine the training process, and reduce overhead to improve accessibility.

## Acknowledgments

This work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government’s Cooperative Research Centres Programme.

## Ethics Considerations

We attest that the research topic, research process, and possible future publication of the research content are ethical. This research aims to facilitate the provable robustness of deep reinforcement learning agents by proposing a novel training method. To the best of our knowledge, the proposed method neither introduces extra risks nor is exploitable for potential adversaries. Moreover, the entire research process involves no experiments with live systems or human/animal participants.

## Open Science

This paper adheres to the principles of open science by ensuring that all research materials, data, and code used in the study are publicly accessible and fully documented. The experimental environments are available in open libraries, promoting transparency and reproducibility of the results. The methodology is described in detail, with the complete code and scripts permanently hosted on <https://zenodo.org/records/14729675> and Github to support replication and validation by other researchers. Furthermore, the study's findings are shared under an open-access license, ensuring that the scientific community and the public can freely access, use, and build upon the work. This commitment to open science promotes collaboration, enhances the credibility of the research, and contributes to the collective advancement of knowledge in the field.

## References

- [1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.
- [2] James L Bander and Chelsea C White III. Markov decision processes with noise-corrupted and delayed state observations. *Journal of the Operational Research Society*, 50(6):660–668, 1999.
- [3] Tamer Başar and Pierre Bernhard. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.
- [4] Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *MLDM*, 2017.
- [5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.
- [6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.
- [7] Andrew Cullen, Paul Montague, Shijie Liu, Sarah Erfani, and Benjamin Rubinstein. Double bubble, toil and trouble: Enhancing certified robustness through transitivity. In *NeurIPS*, 2022.
- [8] Krishnamurthy Dj Dvijotham, Jamie Hayes, Borja Balle, Zico Kolter, Chongli Qin, Andras Gyorgy, Kai Xiao, Sven Gowal, and Pushmeet Kohli. A framework for robustness certification of smoothed classifiers using f-divergences. In *ICLR*, 2020.
- [9] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.
- [10] Raul Fernandez-Fernandez, Juan G Victores, and Carlos Balaguer. Deep robot sketching: an application of deep q-learning networks for human-like sketching. *Cognitive Systems Research*, 81:57–63, 2023.
- [11] Marc Fischer, Maximilian Baader, and Martin Vechev. Certified defense to image transformations via randomized smoothing. In *NeurIPS*, 2020.
- [12] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *ICLR*, 2020.
- [13] Wenbo Guo, Xian Wu, Lun Wang, Xinyu Xing, and Dawn Song. {PATROL}: Provable defense against adversarial policy in two-player games. In *USENIX Security*, 2023.
- [14] Zhongkai Hao, Chengyang Ying, Yinpeng Dong, Hang Su, Jian Song, and Jun Zhu. Gsmooth: Certified robustness against semantic transformations via generalized randomized smoothing. In *ICML*, 2022.
- [15] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Fast bellman updates for robust mdps. In *ICML*, 2018.
- [16] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *ICLR*, 2017.
- [17] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João GM Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

- [18] Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. In *NeurIPS*, 2021.
- [19] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. In *NeurIPS*, 2020.
- [20] Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. In *NeurIPS*, 2020.
- [21] Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *AAAI*.
- [22] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *IEEE Security and Privacy Workshops*, 2018.
- [23] Aounon Kumar, Alexander Levine, and Soheil Feizi. Policy smoothing for provably robust reinforcement learning. In *ICLR*, 2022.
- [24] Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. In *NeurIPS*, 2020.
- [25] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *ICML*, 2020.
- [26] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE SP*, 2019.
- [27] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.
- [28] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *NeurIPS*, 2019.
- [29] Jialian Li, Tongzheng Ren, Dong Yan, Hang Su, and Jun Zhu. Policy learning for robust markov decision process with a mismatched generative model. In *AAAI*, 2022.
- [30] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *CCS*, 2021.
- [31] Linyi Li, Jiawei Zhang, Tao Xie, and Bo Li. Double sampling randomized smoothing. In *ICML*, 2022.
- [32] Xiuju Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. In *ICNLP*, 2017.
- [33] Jieyu Lin, Kristina Dzevaroska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. On the robustness of cooperative multi-agent reinforcement learning. In *IEEE Security and Privacy Workshops*, 2020.
- [34] Xiangyu Liu, Souradip Chakraborty, Yanchao Sun, and Furong Huang. Rethinking adversarial policies: A generalized attack formulation and provable defense in rl. In *ICLR*, 2024.
- [35] Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. On the robustness of safe reinforcement learning under observational perturbations. In *ICLR*, 2023.
- [36] Björn Lütjens, Michael Everett, and Jonathan P How. Certified adversarial robustness for deep reinforcement learning. In *CoRL*, 2020.
- [37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [39] Ronghui Mu, Leandro Soriano Marcolino, Yanghao Zhang, Tianle Zhang, Xiaowei Huang, and Wenjie Ruan. Reward certification for policy smoothed reinforcement learning. In *AAAI*, 2024.
- [40] Arnab Nilim and Laurent Ghaoui. Robustness in markov decision problems with uncertain transition matrices. In *NeurIPS*, 2003.
- [41] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommanan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *AAMAS*, 2018.
- [42] Oren Peer, Chen Tessler, Nadav Merlis, and Ron Meir. Ensemble bootstrapping for q-learning. In *ICML*, 2021.
- [43] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *ICML*.
- [44] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *ICLR*, 2018.



- [45] Aurko Roy, Huan Xu, and Sebastian Pokutta. Reinforcement learning under model mismatch. In *NeurIPS*, 2017.
- [46] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*, 2019.
- [47] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. In *NeurIPS*, 2020.
- [48] Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. Deep reinforcement learning with robust and smooth policy. In *ICML*, 2020.
- [49] Youwei Shu, Xi Xiao, Derui Wang, Yuxin Cao, Siji Chen, Jason Xue, Linyi Li, and Bo Li. Effects of exponential Gaussian distribution on (Double sampling) randomized smoothing. In *ICML*, 2024.
- [50] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.
- [51] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *CoRL*, 2022.
- [52] Peter Šukeník, Aleksei Kuvshinov, and Stephan Günnemann. Intriguing properties of input-dependent randomized smoothing. In *ICML*, 2022.
- [53] Chung-En Sun, Sicun Gao, and Tsui-Wei Weng. Breaking the barrier: Enhanced utility and robustness in smoothed DRL agents. In *ICML*, 2024.
- [54] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *AAAI*, 2020.
- [55] Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. Certifiably robust policy learning against adversarial multi-agent communication. In *ICLR*, 2023.
- [56] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.
- [57] Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations. In *arXiv preprint arXiv:2008.01825*, 2020.
- [58] Derui Wang, Minhui Xue, Bo Li, Seyit Camtepe, and Liming Zhu. Provably unlearnable data examples. In *NDSS*, 2025.
- [59] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
- [60] Fan Wu, Linyi Li, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. Crop: Certifying robust policies for reinforcement learning through functional smoothing. In *ICLR*, 2022.
- [61] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *ICML*, 2020.
- [62] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *ICAIF*, 2020.
- [63] Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. Rorl: Robust offline reinforcement learning via conservative smoothing. In *NeurIPS*, 2022.
- [64] Wenhao Yang, Liangyu Zhang, and Zhihua Zhang. Toward theoretical understandings of robust markov decision processes: Sample complexity and asymptotics. *The Annals of Statistics*, 50(6):3223–3248, 2022.
- [65] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *ICLR*, 2020.
- [66] Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. In *NeurIPS*, 2020.
- [67] Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *ICLR*, 2020.
- [68] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In *NeurIPS*, 2020.
- [69] Zhengqing Zhou, Zhengyuan Zhou, Qinxun Bai, Linhai Qiu, Jose Blanchet, and Peter Glynn. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In *AISTATS*, 2021.

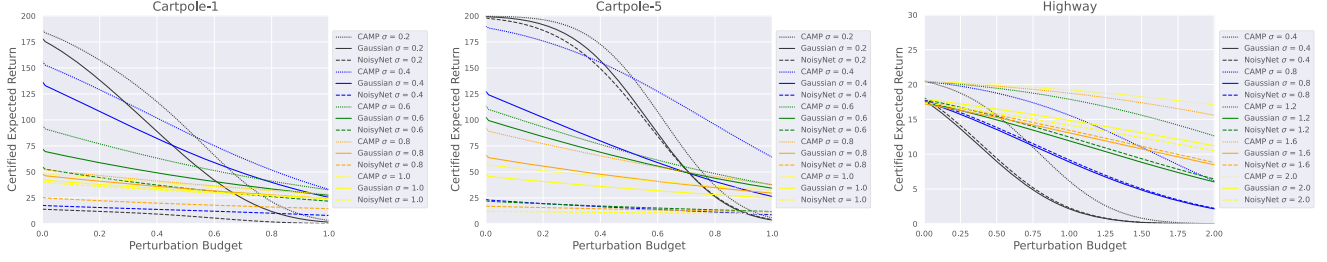


Figure 8: Full certification results on Cartpole-1, Cartpole-5, and Highway.

## A Proofs

**Theorem 1 (Change of variable).** Given a target expected return threshold  $\xi$  for the randomized policy, let the perturbed trajectory be  $\mathbf{z}' = \mathbf{z} + \Delta$  and define  $P_{\pi}^{\mathbf{z}}(C) := \Pr[F_{\pi}(\mathbf{z}) \geq C]$ . Let  $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_m\}$  represent a set of sampled and sorted values of  $F_{\pi}(\mathbf{z})$  such that  $\mathbf{r}_1 \leq \mathbf{r}_2 \leq \dots \leq \mathbf{r}_m$ . If  $\xi/\mathbf{r}_1 \leq P_{\pi}^{\mathbf{z}}(\mathbf{r}_1)$  and

$$\|\Delta\|_2 \leq \sigma \left[ \Phi^{-1}(P_{\pi}^{\mathbf{z}}(\mathbf{r}_1)) - \Phi^{-1}(\xi/\mathbf{r}_1) \right], \quad (8)$$

then  $\mathbb{E}[F_{\pi}(\mathbf{z}')] \geq \xi$ .

*Proof.* By the lower bound on  $\mathbb{E}[F_{\pi}(\mathbf{z})]$  and algebra,

$$\begin{aligned} \mathbb{E}[F_{\pi}(\mathbf{z}')] &\geq \mathbf{r}_1 \cdot \Phi \left( \Phi^{-1} \left( P_{\pi}^{\mathbf{z}}(\mathbf{r}_1) \right) - \frac{\tau}{\sigma} \right) \\ &\quad + \sum_{i=2}^m (\mathbf{r}_i - \mathbf{r}_{i-1}) \cdot \Phi \left( \Phi^{-1} \left( P_{\pi}^{\mathbf{z}}(\mathbf{r}_i) \right) - \frac{\tau}{\sigma} \right), \\ &\geq \mathbf{r}_1 \cdot \Phi \left( \Phi^{-1} \left( P_{\pi}^{\mathbf{z}}(\mathbf{r}_1) \right) - \frac{\tau}{\sigma} \right) \\ &\quad \forall \|\Delta\|_2 \leq \tau. \end{aligned} \quad (21)$$

Therefore to make  $\mathbb{E}[F_{\pi}(\mathbf{z}')] \geq \xi$ , it is sufficient if

$$\begin{aligned} \mathbf{r}_1 \cdot \Phi \left( \Phi^{-1} \left( P_{\pi}^{\mathbf{z}}(\mathbf{r}_1) \right) - \frac{\tau}{\sigma} \right) &\geq \xi, \\ \Leftrightarrow \tau &\leq \sigma \left[ \Phi^{-1} \left( P_{\pi}^{\mathbf{z}}(\mathbf{r}_1) \right) - \Phi^{-1}(\xi/\mathbf{r}_1) \right]. \end{aligned} \quad (22)$$

□

**Theorem 2 (Correlation between CDF and expectation).** Given two random variables  $X \geq 0$  and  $Y \geq 0$ , let their expectations be  $\mathbb{E}[X]$  and  $\mathbb{E}[Y]$ , respectively. The following inequality holds if and only if  $\mathbb{E}[X] \leq \mathbb{E}[Y]$ :

$$\int_0^{+\infty} [\Psi_X(C) - \Psi_Y(C)] dC \geq 0, \quad (9)$$

where  $\Psi_X(C)$  and  $\Psi_Y(C)$  are the CDFs of  $X$  and  $Y$ , respectively.

*Proof.* Since  $\Psi_X(C)$  and  $\Psi_Y(C)$  are CDFs, by definition:

$$\begin{aligned} \Psi_X(C) &= 1 - \Pr[X \geq C] \\ &= 1 - \int_C^{+\infty} f_X(t) dt. \end{aligned} \quad (23)$$

Note that given  $X \geq 0$  and  $Y \geq 0$ ,

$$\begin{aligned} \int_0^{+\infty} [1 - \Psi_X(C)] dC &= \int_0^{+\infty} \Pr[X \geq C] dC \\ &= \int_0^{+\infty} \int_C^{+\infty} f_X(t) dt dC \\ &= \int_0^{+\infty} \int_0^t f_X(t) dx dt \\ &= \int_0^{+\infty} t f_X(t) dt. \end{aligned} \quad (24)$$

Let  $t = C$  and thus  $dt = dC$ , there is

$$\begin{aligned} \int_0^{+\infty} (1 - \Psi_X(C)) dC &= \int_0^{+\infty} t f_X(t) dt \\ &= \int_0^{+\infty} C f_X(C) dC \\ &= \mathbb{E}[X]. \end{aligned} \quad (25)$$

In the same spirit, we can obtain

$$\mathbb{E}[Y] = \int_0^{+\infty} [1 - \Psi_Y(C)] dC. \quad (26)$$

Therefore we have

$$\begin{aligned} \mathbb{E}[X] &\leq \mathbb{E}[Y] \\ \Leftrightarrow \int_0^{+\infty} [1 - \Psi_X(C)] dC &\leq \int_0^{+\infty} [1 - \Psi_Y(C)] dC \\ \Leftrightarrow \int_0^{+\infty} [\Psi_Y(C) - \Psi_X(C)] dC &\geq 0. \end{aligned} \quad (27)$$

□

**Lemma 1 (Lipschitz continuity of smoothed return function).** For any measurable function  $F_{\pi} : \mathbf{z} \in (\mathbb{S} \times \mathbb{A} \times \mathbb{R}^d)^T \rightarrow [A, B]$ , let  $\mathbb{E}[F_{\pi}(\mathbf{z})] = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} F_{\pi}(\mathbf{z} + \epsilon)$ . There is  $\mathbf{z} \rightarrow \mathbb{E}[F_{\pi}(\mathbf{z})]$  is  $\frac{(B-A)}{\sigma} \sqrt{2/\pi}$ -Lipschitz.

*Proof.* See proofs of Lemma 2 in Fan *et al.* [60]. □

**Theorem 3 (Soft certified radius).** Given a target expected return  $\xi \leq \mathbb{E}[F_{\pi}(\mathbf{z})]$  where  $F_{\pi} : \mathbf{z} \in (\mathbb{S} \times \mathbb{A} \times \mathbb{R}^d)^T \rightarrow [A, B]$ , suppose an adversary perturbs the observed states by applying  $\Delta = (\delta_0, \delta_1, \dots, \delta_{T-1})$ . The target expected return will not drop below  $\xi$  if the perturbations satisfy:

$$\|\Delta\|_2 \leq \sigma \left[ \Phi^{-1} \left( \frac{\mathbb{E}[F_{\pi}(\mathbf{z})] - A}{B - A} \right) - \Phi^{-1} \left( \frac{\xi - A}{B - A} \right) \right]. \quad (12)$$

Refer to the proof in the Section 4.2.

**Theorem 4 (Local certified radius).** Let  $l_i$  and  $u_i$  denote the lower and upper bounds of the expected action-value function at step  $i$ , respectively. Let the perturbations from step  $t$  to step  $T-1$  be  $\{\delta_i\}_{i=t}^{T-1}$ . Under a greedy policy, the optimal reward at step  $t$  is obtained by taking action  $a_i^{(1)} = \arg \max_{a_i} \bar{Q}_{\pi^*}(s_i, a_i)$ . The optimal expected return from step  $t$  to  $T-1$  will not be reduced if the local perturbation  $\delta_i$  at each step  $i$  satisfies:

$$\|\delta_i\|_2 \leq \frac{\sigma}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(2)}) - l_i}{u_i - l_i} \right) \right], \quad (16)$$

where  $a_i^{(2)}$  is the runner-up action  $a_i^{(2)} = \arg \max_{a_i: a_i \neq a_i^{(1)}} \bar{Q}_{\pi^*}(s_i, a_i)$ .

*Proof.* According to the Lemma 1 of Salman et al. [46],  $\mathbb{E}_{\epsilon_t} Q_{\pi}(s_t + \epsilon_t, a_t)$  is  $\sqrt{2/\pi} \frac{u_i - l_i}{\sigma}$ -Lipschitz and  $\Phi^{-1} \left( \frac{\mathbb{E}_{\epsilon_t} Q_{\pi}(s_t + \epsilon_t, a_t) - l_i}{u_i - l_i} \right)$  is 1-Lipschitz. Since  $\bar{Q}_{\pi^*}(s_t, a_t) = \max_{\pi} \mathbb{E}_{\epsilon_t} Q_{\pi}(s_t + \epsilon_t, a_t)$ , without loss of generality,  $\bar{Q}_{\pi^*}(s_t, a_t)$  has the same Lipschitz continuity. Let  $a_i^{(1)} = \arg \max_{a_i} \bar{Q}_{\pi^*}(s_i, a_i)$  and  $a_i^{(2)} = \arg \max_{a_i: a_i \neq a_i^{(1)}} \bar{Q}_{\pi^*}(s_i, a_i)$ . Based on Lemma 1, there are

$$\Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \leq \|\delta_i\|_2, \quad (28)$$

and for all  $a'_i: a'_i \neq a_i^{(1)}$ ,

$$\Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a'_i) - l_i}{u_i - l_i} \right) - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a'_i) - l_i}{u_i - l_i} \right) \leq \|\delta_i\|_2. \quad (29)$$

If the perturbation  $\delta_i$  satisfies

$$\|\delta_i\|_2 \leq \frac{\sigma}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a'_i) - l_i}{u_i - l_i} \right) \right], \quad (30)$$

we have:

$$\begin{aligned} & \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \\ & \stackrel{(a)}{\geq} \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \\ & - \frac{1}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \right. \\ & \left. - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a'_i) - l_i}{u_i - l_i} \right) \right] \end{aligned} \quad (31)$$

and

$$\begin{aligned} & \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a'_i) - l_i}{u_i - l_i} \right) \\ & \stackrel{(b)}{\leq} \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a'_i) - l_i}{u_i - l_i} \right) \\ & + \frac{1}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \right. \\ & \left. - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \epsilon_i, a_i^{(2)}) - l_i}{u_i - l_i} \right) \right] \\ & \stackrel{(c)}{\leq} \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(2)}) - l_i}{u_i - l_i} \right) \\ & + \frac{1}{2} \left[ \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \right. \\ & \left. - \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i, a_i^{(2)}) - l_i}{u_i - l_i} \right) \right] \end{aligned} \quad (32)$$

Herein, (a) and (b) hold according to Equation 28, Equation 29, and Condition 30. (c) is due to  $\bar{Q}_{\pi^*}(s_i, a'_i) \leq \bar{Q}_{\pi^*}(s_i, a_i^{(2)})$  and the monotonicity of  $\Phi^{-1}(\cdot)$ .

Therefore, we have

$$\Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a_i^{(1)}) - l_i}{u_i - l_i} \right) \geq \Phi^{-1} \left( \frac{\bar{Q}_{\pi^*}(s_i + \delta_i, a'_i) - l_i}{u_i - l_i} \right) \quad (33)$$

$$\iff \bar{Q}_{\pi^*}(s_i + \delta_i, a_i^{(1)}) \geq \bar{Q}_{\pi^*}(s_i + \delta_i, a'_i), \forall a'_i \neq a_i^{(1)}.$$

This means that the top-1 action recommended by the policy retains the highest expected return at step  $t$ . If each perturbation in  $\{\delta_i\}_{i=t}^{T-1}$  follows the above condition, the optimal expected return will not be decreased.  $\square$

## B Experiment Settings

### B.1 Hyper-Parameters and Settings

**Cartpole.** For training on Cartpole-1 and Cartpole-5 using CAMP, we employ a buffer size of 100k, a total of 500k training steps, a batch size of 1024, and a learning rate of  $5 \times 10^{-5}$ . Training begins after 1000 burn-in steps. The training frequency is set to 256, with each step involving 128 gradient updates. The target Q-network is updated every 10 steps with a Polyak averaging rate of 1. An  $\epsilon$ -greedy policy is used during training, where  $\epsilon$  follows a linear schedule starting from 1 and decreasing to 0 over the first 16% of training steps. Validation occurs every 2000 training steps, with the validation return averaged over 10 validation episodes.

**Highway.** A buffer size of 15k is used for Highway. The training lasts for 1000k steps, including 1k burn-in steps.  $\epsilon$  follows a linear schedule starting from 1 and decreasing to 0.05 over the first 10% of training steps. The other hyper-parameters are the same as those in the Cartpole environments.

**Atari.** The reference network training involves a buffer size of 10k, a total of 10 million training steps, a batch size of 32, and a learning rate of  $1 \times 10^{-4}$ . Training begins after a burn-in period of 100k steps. The training frequency is set to 4, with

Table 4: Network Architectures

MLP	Nature CNN
<i>input dim</i>	$84 \times 84$
Linear $\text{input dim} \times 256$	$\text{Conv2d}(4, 32, 8, \text{stride} = 4)$
ReLU	ReLU
Linear $256 \times 256$	$\text{Conv2d}(32, 64, 4, \text{stride} = 2)$
ReLU	ReLU
Linear $256 \times \text{action dim}$	$\text{Conv2d}(64, 64, 3, \text{stride} = 1)$
-	ReLU
-	Flatten()
-	Linear $3136 \times 512$
-	ReLU
-	Linear $512 \times \text{action dim}$

each training step involving a single gradient step. The target Q-network is updated every 1000 steps with a Polyak rate of 1. The  $\epsilon$ -greedy policy's  $\epsilon$  value follows a linear schedule, starting at 1 and gradually decreasing to 0.01 over the first 10% of the training steps. Validation occurs every 100k training steps, with the validation return averaged over 100 episodes. To prevent excessively long gameplay, the frame number is capped at 250 for all games. The primary Q-networks are trained for 1 million steps using a 0.01-greedy policy, with validation performed every 1,000 training steps. All other settings remain consistent with those used for training reference policies.

We also observed a discrepancy between the certification and empirical robustness results of our Gaussian baseline and those reported in the PS paper. The discrepancy may stem from differences in training settings. First, our method is implemented using the Clean-RL library [17], whereas the other implementation is based on Stable Baselines 3. Second, during Atari game training, we normalize the observed pixel values to the range  $[0, 1]$ , while the other implementation uses observations in the range  $[0, 255]$ .

## B.2 Additional Results

The full certification results in Cartpole-1/5 and Highway across different  $\sigma$  setups are presented in Figure 8. It is evident that CAMP consistently outperforms the baseline methods across all  $\sigma$  levels, with the most significant improvement occurring in Cartpole-5 when  $\sigma = 0.4$ .

## B.3 Model Architectures

We list the architectures of the Q-networks used in this paper. For classic control problems, we use MLP following previous papers. Nature CNN is used elsewhere for Atari games. Table 4 summarizes the architectures of these two networks.

## C Additional Algorithmic Details

We provide the details of the empirical attacks used to evaluate the robustness of the trained policies in Algorithm 2.  $\text{Perturb\_Step}(s'_t; c, l_{CE}, \alpha)$  represents either a single PGD or APGD step as defined in their original literature (*i.e.*, PGD

### Algorithm 2: Empirical Attack

**Input:** Budget  $\tau$ , agent with Q-network  $\pi$ , attack step size  $\alpha$ , multiplier  $\beta$ , episode length  $T$ , Q value gap  $q$ , environment  $E$  with state transition function  $\mathcal{T}$  and reward function  $\mathcal{R}$ .

```

1 Initialize currently available budget  $c = \tau$ 
2 Initialize  $s_0$ 
3 Initialize  $\Delta = \emptyset$ 
4 for  $t \in \{0, 1, 2, \dots, T-1\}$  do
5    $a_t^* \leftarrow \arg \max_a \pi(s_t)_a$ 
6    $Q_t^* \leftarrow \pi(s_t)_{a_t^*}$ 
7    $s_t^* \leftarrow s_t$ 
8    $\text{step\_count} \leftarrow \lfloor \beta * c / \alpha \rfloor$ 
9   for  $a' \in \mathbb{A}, a' \neq a_t^* \text{ and } |a' - a_t| \geq q$  do
10     $s'_t \leftarrow s_t$ 
11    for  $i \in \{1, 2, \dots, \text{step\_count}\}$  do
12       $a_{\text{predicted}} \leftarrow \arg \max_a \pi(s'_t)_a$ 
13      if  $a_{\text{predicted}} == a'$  then
14        if  $\pi(s_t)_{a_{\text{predicted}}} < Q_t^*$  then
15           $Q_t^* \leftarrow \pi(s_t)_{a_{\text{predicted}}}$ 
16           $s_t^* \leftarrow s'_t$ 
17          break
18     $l_{CE} \leftarrow CE(\pi(s'_t), \text{One\_Hot}(a'))$ 
19     $s'_t \leftarrow \text{Perturb\_Step}(s'_t; c, l_{CE}, \alpha)$ 
20   $c \leftarrow \sqrt{c^2 - \|s_t^* - s_t\|_2^2}$ 
21  if  $c < 0$  then
22    break
23   $\Delta \leftarrow \Delta \cup \{s_t^* - s_t\}$ 
24   $a_t \leftarrow \arg \max_a \pi(s_t^*)_a$ 
25   $s_{t+1}, d_t, r_{t+1} \leftarrow \mathcal{T}(s_t, a_t), \mathcal{R}(s_t, a_t)$ 

```

**Output:** Observation perturbations  $\Delta$

step [37], or Line 7-Line 16, Algorithm 1 in APGD [6]).  $l_{CE}$  is the Cross-Entropy loss between predicted logits and the one-hot encoding of the target action.

We slightly modify the APGD attack to attack DRL agents. Specifically, we omit random initialization of adversarial examples since it quickly runs out of perturbation budget and leads to unsuccessful attacks. Moreover, we decrease the initial step size  $\alpha$  to  $0.05 * \tau$  ( $0.5 * \tau$  in Bank Heist), such that the budget can be better preserved and allocated across the episode. In both attacks, the step count at each time step is calculated as  $\beta * c / \alpha$ , where  $c$  is the perturbation budget available at the current step and  $\beta = 2$ . On the side of PGD, we apply the normalized gradients rather than gradient signs in each attack step. The step size  $\alpha$  in PGD is 0.01.

APGD halves its step size during the attack process, allowing it to meet the attack criteria (*i.e.*, causing observations to be classified into target actions by the Q-network) with smaller perturbations at each step. As a result, APGD may induce a more significant drop in the average return. Nevertheless, across all environments, CAMP agents consistently demonstrate more robust performance compared to baseline agents.