CERTIFIED DIFFUSION SMOOTHING FOR ROBUST REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Reinforcement Learning (DRL) is known to be vulnerable to small adversarial perturbations in the input space, which limits its deployment in real-world applications. While many empirical studies have sought to improve its robustness against adversarial attacks, these methods typically lack formal robustness guarantees. Existing certified defenses, despite providing certification criteria, often suffer from low robust rewards and weak robustness guarantees. In this work, we introduce a certifiably robust reinforcement learning framework that leverages the diffusion process to incorporate the objective of randomized smoothing directly into policy training. We propose Certified Diffusion Smoothing (CDS), a novel method that trains a smoothed agent with stronger certified robustness guarantees and higher robust rewards. We evaluate CDS under multiple certifications and adversarial attacks in standard RL environments, including MuJoCo and Atari. Across these benchmarks, CDS consistently outperforms previous smoothed agents by an average factor of 1.5×–3.2× and robustly trained agents by an average factor of 2.8×. Our code is available at `https://anonymous.4open.science/r/CDS_DRL`.

1 INTRODUCTION

Deep Reinforcement Learning (DRL) has achieved remarkable success in solving complex sequential decision-making tasks Kalashnikov et al. (2018); Rudin et al. (2022); Kaiser et al. (2019). However, DRL policies are notoriously vulnerable to perturbations and adversarial attacks, where imperceptible changes to the state observations can catastrophically mislead the actions Szegedy et al. (2013); Chakraborty et al. (2018); Nie et al. (2024). While empirical defenses, such as adversarial training Gleave et al. (2019); Sun et al. (2020), offer some degree of resilience, they fail against adaptive or strong iterative attacks. This necessitates the development of certified defenses, which provide theoretically provable robustness guarantees regardless of the attack strategy.

A classic and widely adopted approach in certified defenses is randomized smoothing Cohen et al. (2019); Wu et al. (2022a); Scholten et al. (2023); Bansal et al. (2022). It converts a given classifier f into a smoothed classifier g, which is characterized by a non-linear Lipschitz property Salman et al. (2019). When queried at a point x, the smoothed classifier g outputs the class that is most likely to be returned by f under isotropic Gaussian perturbations of its inputs Salman et al. (2020). If there is a sufficiently large margin in probabilities between the top-1 and runner-up predictions, the smoothed classifier is guaranteed to be robust to ℓ_2-norm bounded adversarial perturbations.

Although randomized smoothing provides theoretical robustness guarantees, its certification quality fundamentally depends on the robustness of the base policy Salman et al. (2020). Existing methods in reinforcement learning typically apply randomized smoothing only at evaluation time, treating it as a post-hoc certification wrapper around a pretrained agent. However, such a strategy does not address the fragility of the original decision process Jeong & Shin (2023). Certification is only meaningful when the policy induces a sufficiently stable smoothed decision boundary in the local perturbation region. When the base policy is highly vulnerable to adversarial or random input perturbations, randomized smoothing may still return weak or vacuous certificates, often accompanied by low robust rewards under attack, thereby limiting its practical value Sun et al. (2024).

To bridge this gap, we propose CDS, a certifiably robust reinforcement learning framework that treats randomized smoothing not merely as an evaluation-time certification wrapper, but as a training-time consistency principle for policy learning. CDS optimizes the policy over Gaussian perturbation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

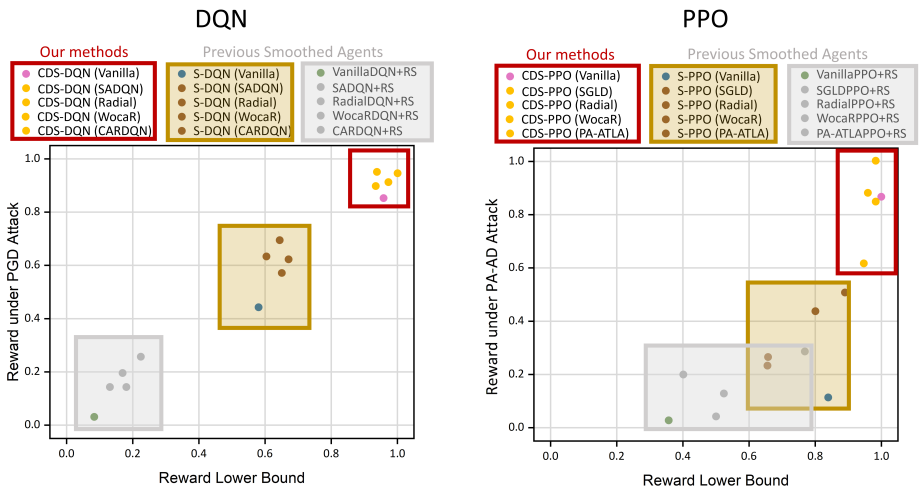


Figure 1: Robust reward and cumulative reward lower bound for smoothed agents. The reported rewards are normalized and averaged across environments. Our CDS (red boxes) agents exhibit improved robustness compared with previous smoothed agents.

neighborhoods used for smoothing, so that policy learning is aligned with the perturbation distribution used for certification. CDS leverages diffusion models to stabilize this training-time smoothing process by aligning the Gaussian parameters of randomized smoothing with those of the diffusion forward process. The reverse process then serves as a denoising prior, steering noisy observations toward locally plausible regions. This design stabilizes the smoothing process during training and reduces policy inconsistency across noisy samples. As summarized in Figure 1, CDS improves both robust reward under adversarial attacks and certified performance measured by the cumulative reward lower bound.

Our contribution can be summarized as follows:

- We propose CDS, a certifiably robust reinforcement learning framework that treats randomized smoothing as a training-time consistency principle. CDS aligns the Gaussian perturbation distribution used for smoothing with the diffusion process, stabilizing the smoothing process during policy optimization.
- We instantiate two CDS-based agents, CDS-DQN and CDS-PPO, with distinct diffusion architectures tailored to discrete and continuous action spaces, respectively, demonstrating the adaptability of the proposed framework across different reinforcement learning settings.
- Extensive experiments show that CDS achieves superior performance across multiple certified robustness metrics and robust rewards under diverse adversarial attacks.

2 Related Work

Adversarial attacks. Adversarial attacks for RL are widely studied in the literature Lin et al. (2017); Pattanaik et al. (2018). These attacks generate perturbed samples by adding an imperceptible perturbation δ to clean observations s, thus misleading DRL agents. The magnitude of δ is quantified by its ℓ_p norm, denoted as $||\delta||_p \leq \varepsilon$. Many adversarial attack methods Gleave et al. (2019); Sun et al. (2020); Sarkar et al. (2022) have been proposed to achieve high attack success rates under white-box or black-box settings. To defend against adversarial attacks, adversarial training Kos & Song (2017); Behzadan & Munir (2017); Liang et al. (2022); Zhang et al. (2023) and regularized RL algorithms that induce smoothness of trained policies Shen et al. (2020); Kobayashi (2022); Massiani et al. (2024); Li et al. (2024) have been widely studied. However, these methods tend to show robustness only to the specific attack they are trained on, and are vulnerable to strong adaptive attacks proposed against these empirical defenses Korkmaz & Brown-Cohen (2023); Korkmaz (2023); Bai et al. (2025).

Certified Defenses. To put an end to this cat-and-mouse game between attackers and defenders Wu et al. (2022a), certified defenses provide theoretical guarantees of robustness against ℓ_p-norm bounded adversarial perturbations. The current state-of-the-art approach to certifying robustness against adversarial perturbations bounded in ℓ_2-norm is randomized smoothing Cohen et al. (2019); Lecuyer et al. (2019); Scholten et al. (2023); Voráček (2024), which is widely studied in image classification. Recently, certified robustness research for reinforcement learning has attracted increasing attention, particularly by accounting for RL's sequential decision-making nature Kumar et al. (2021); Wu et al. (2022a;b); Sun et al. (2024). Denoised smoothing further emerged as an enhanced variant of randomized smoothing Salman et al. (2020); Carlini et al. (2023); Sun et al. (2024). Such methods adopt a denoising mechanism to mitigate the adverse effect of noisy states on clean rewards, while improving the effectiveness of smoothing and thereby enhancing certified robustness.

Diffusion Models in Reinforcement Learning. Diffusion models are originally motivated by non-equilibrium thermodynamics Sohl-Dickstein et al. (2015); Ho et al. (2020), defining data generation as an iterative denoising process. Beyond being powerful function approximators, they have shown impressive performance on generative modeling tasks Karras et al. (2022); Jing et al. (2022); Song et al. (2021) and have been applied to various tasks such as image inpainting, super-resolution, and even text-to-image synthesis Bansal et al. (2023); Dhariwal & Nichol (2021); Rombach et al. (2022); Saharia et al. (2022). Recently, diffusion models have drawn increasing attention in RL research, functioning as planners for high-reward trajectory generation Ni et al. (2023); Liang et al. (2023); Hu et al. (2023), as policies for rich multimodal behavior representation Wang et al. (2023); Chi et al. (2025); Hansen-Estruch et al. (2023), and as data synthesizers alleviating data scarcity Ho et al. (2020); Sinha et al. (2022); Rigter et al. (2023). However, diffusion-based approaches for improving RL robustness remain limited. AD-RRL Foffano et al. (2025) trains robust policies by guiding a diffusion trajectory model toward worst-case rollouts under dynamics uncertainty. ADG Liu et al. (2025) enhances the robustness of offline RL through a diffusion model to guide the recovery of corrupted rewards. To the best of our knowledge, our work is the first to provide certified robustness guarantees for RL with diffusion models.

3 PRELIMINARIES

Markov decision process (MDP). A Markov decision process is defined by a tuple $(\mathcal{S}, \mathcal{A}, r, \mathbb{P}, \gamma, \mu_0)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ stands for the transition dynamics where $\Delta(\mathcal{S})$ is the probability space over $\mathcal{S}$, $\gamma \in [0, 1)$ represents the discount factor, and $\mu_0 \in \Delta(\mathcal{S})$ is the initial state distribution. Given a state $s \in \mathcal{S}$, the probability of selecting an action $a \in \mathcal{A}$ is defined as $\pi(a|s) := \Pr(A_t = a \mid S_t = s)$. In this paper, we consider compact continuous state spaces $\mathcal{S} \subset \mathbb{R}^d$, with finite action spaces for DQN agents and continuous action spaces for PPO agents. Given an MDP, we define the state value function $V^\pi(s) = \mathbb{E}_{\pi,\mathbb{P}}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s]$ and the Q (or action-value) function $Q^\pi(s, a) = \mathbb{E}_{\pi,\mathbb{P}}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a]$ for every policy π.

Denoising Diffusion Probabilistic Models (DDPM). A DDPM is a type of generative model that learns to denoise noisy samples. The forward process of DDPM constructs a Markov chain $\{x_0, \ldots, x_T\}$ from a clean observation to pure noise. At each time step t, it adds Gaussian noise to the previous state $x_t = \sqrt{1 - \beta_t}\, x_{t-1} + \mathcal{N}(0, \beta_t I)$, where $0 < \beta_1 < \cdots < \beta_T < 1$ control the variance of the diffusion process. Denoting $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t} (1 - \beta_s)$, the forward noising diffusion process is $q(x_t \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\, x_0,\, (1 - \bar{\alpha}_t) I)$. The reverse process of DDPM aims to gradually remove noise from x_t and recover the clean sample x_0 by learning a parameterized Markov chain that approximates the true posterior. Specifically, the reverse diffusion process is defined as $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t),\, \Sigma_t)$, where $\mu_\theta(x_t, t)$ denotes the mean predicted by a neural network parameterized by θ, and Σ_t is a fixed or learned variance.

4 METHODOLOGY

In this section, we present Certified Diffusion Smoothing (CDS), a certifiably robust reinforcement learning framework. Our framework is instantiated in two certifiably robust reinforcement learning agents: CDS-DQN for discrete action spaces and CDS-PPO for continuous action spaces. In the

following, we first introduce the general formulation of CDS, and then describe its training and inference procedures for the two settings.

4.1 CDS-DQN (Certified Diffusion Smoothed - Deep Q Network)

We next introduce CDS-DQN by presenting its diffusion-smoothed policy and loss function, followed by certification procedures.

Training and Loss Function During training, CDS-DQN consists of a trajectory collection stage and a parameter update stage. In the trajectory collection stage, given the current state s_t, we first generate m Gaussian-perturbed copies $\tilde{s}_t^{(i)} = s_t + \epsilon_t^{(i)}$, where $\epsilon_t^{(i)} \sim \mathcal{N}(0, \sigma^2 I)$. Specifically, by aligning the parameters of the Gaussian perturbation distributions, CDS-DQN uses the forward diffusion process to implement this Gaussian sampling step. We provide a detailed discussion in Section 4.1. Then the perturbed states are passed through the reverse process of diffusion model $D_\phi(\cdot)$ to obtain denoised states $\hat{s}_t^{(i)} = D_\phi(\tilde{s}_t^{(i)})$. By removing perturbations and steering noisy samples toward the clean state distribution, reverse diffusion enables CDS to perform smoothing over more stable state representations. We define the smoothed action as the action most frequently selected by the base Q-network over the local denoised neighborhood,

$$a_t^{\text{smooth}} = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{m} \mathbf{1}\left[a = \arg\max_{a' \in \mathcal{A}} Q\left(\hat{s}_t^{(i)}, a'\right)\right]. \tag{1}$$

Here, Q_θ is the pretrained Q-network with parameters θ, and $\mathcal{A}$ is the discrete action space. The agent then follows an ε-greedy smoothed policy:

$$a_t = \begin{cases} a_t^{\text{smooth}}, & \text{with probability } 1-\varepsilon, \\ \text{Random Action}, & \text{with probability } \varepsilon. \end{cases} \tag{2}$$

The resulting transitions (s_t, a_t, r_t, s_{t+1}) are stored in the replay buffer for subsequent training. In the training stage, CDS-DQN updates the Q-network using the temporal-difference objective

$$\mathcal{L}_{\text{Q}} = \left(r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a') - Q_\theta(D_\phi(\tilde{s}_t), a_t)\right)^2, \tag{3}$$

where γ is the discount factor, Q_{θ^-} is the target network, and $\tilde{s}_t = s_t + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$ denotes the perturbed state. Subsequently, the diffusion model is trained to remove Gaussian perturbations from noisy states through the standard denoising objective

$$\mathcal{L}_{\text{D}} = \mathbb{E}_{t,s,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\phi(\sqrt{\bar{\alpha}_t}s + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]. \tag{4}$$

Therefore, the diffusion model and the Q-network are jointly optimized by the following overall objective: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{D}} + \lambda_2 \mathcal{L}_{\text{Q}}$, where λ_1 and λ_2 are balancing coefficients.

Gaussian Alignment Between Randomized Smoothing and Diffusion. Following previous work in supervised learning Carlini et al. (2023), we use the probabilistic correspondence between randomized smoothing and diffusion models: randomized smoothing adds Gaussian noise to the input, while the diffusion forward process defines Gaussian marginals with timestep-dependent parameters. By matching these Gaussian parameterizations, we realize the smoothing perturbation through the forward diffusion process and apply reverse diffusion within the aligned Gaussian neighborhood to map perturbed states to more stable representations. Specifically, randomized smoothing requires a data point augmented with additive Gaussian noise $x_{rs} \sim \mathcal{N}(x, \sigma^2 \mathbf{I})$, whereas the DDPM marginal noise model is $x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x, (1-\bar{\alpha}_t)\mathbf{I})$. Scaling x_{rs} by $\sqrt{\bar{\alpha}_t}$ and equating the variances yields

$$\sigma^2 = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}. \tag{5}$$

Thus, for a given smoothing noise level σ, we first find the nearest diffusion timestep $t^\star$ such that $\sigma^2 \approx \frac{1-\bar{\alpha}_{t^\star}}{\bar{\alpha}_{t^\star}}$ and obtain $\tilde{s}$ through the forward diffusion process $\sqrt{\bar{\alpha}_{t^\star}}\left(x + \mathcal{N}(0, \sigma^2 \mathbf{I})\right)$. We then use $t^\star$ as the starting point of the reverse diffusion process $D_\phi(\tilde{s})$.

Action-Level Certification. For RL with a discrete action space, we construct an action-level certification. We define *Action Probability Function* $f : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Given a specific state $s_t \in \mathcal{S}$ and an action $a \in \mathcal{A}$, the function $f(s_t, a)$ is constructed through a Monte Carlo sampling process Cohen et al. (2019):

$$f(s_t, a) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[a = \arg\max_{a' \in \mathcal{A}} Q\Big(D(s_t + \delta^{(i)}), a'\Big)\right], \qquad \delta^{(i)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{6}$$

Based on this action probability function, the certified radius is given by:

$$R_t = \frac{\sigma}{2} \left(\Phi^{-1}(f(s_t, a_1)) - \Phi^{-1}(f(s_t, a_2))\right), \tag{7}$$

a_1 is the most probable action, and a_2 is the runner-up action. R_t is the certified radius at time t, Φ is the CDF of the standard normal distribution, and σ is the smoothing variance. As long as the ℓ_2-norm of the perturbation is bounded by R_t, the smoothed policy is guaranteed to remain consistent: $\tilde{\pi}_a(s_t + \delta) = \tilde{\pi}_a(s_t)$. The proof is provided in Appendix A.1.

Value-Level Certification. Complementing action-level certification, we further certify robustness from the value perspective by focusing on inter-action value gaps. For each state s_t, we compute the smoothed Q value:

$$\hat{Q}^{\pi}(s_t, a) = \frac{1}{N} \sum_{j=1}^{N} \mathrm{clip}_{[0,1]} \left(\frac{Q^{\pi}\big(D(s_t + \delta^{(j)}), a\big) - Q^{(t)}_{\min}}{Q^{(t)}_{\max} - Q^{(t)}_{\min}} \right). \tag{8}$$

To satisfy the independence assumptions required by the theoretical analysis, $Q^{(t)}_{\max}$ and $Q^{(t)}_{\min}$ are estimated from an independent sample set. We clip each normalized sample before averaging so that $\hat{Q}^{\pi}(s_t, a) \in [0, 1]$, which keeps the Gaussian quantile transform well-defined. Based on the normalized smoothed values, the certified radius is given by

$$R_t = \frac{\sigma}{2} \left(\Phi^{-1}\Big(\hat{Q}^{\pi}(s_t, a_1)\Big) - \Phi^{-1}\Big(\hat{Q}^{\pi}(s_t, a_2)\Big)\right), \tag{9}$$

where $a_1 = \arg\max_{a \in \mathcal{A}} \hat{Q}^{\pi}(s_t, a)$ and $a_2 = \arg\max_{a \in \mathcal{A} \setminus \{a_1\}} \hat{Q}^{\pi}(s_t, a)$ are the top and runner-up actions, respectively. The proof is provided in Appendix A.2.

4.2 CDS-PPO (Certified Diffusion Smoothed - Proximal Policy Optimization)

We next introduce CDS-PPO by presenting its diffusion-smoothed policy and loss function, followed by certification procedures.

Conditional Diffusion and Median Smoothing For continuous control, policies are expected to produce semantically consistent actions across adjacent states. We therefore instantiate CDS-PPO with a conditional diffusion model, which uses temporal state-action context to obtain a more stable state representation. Specifically, let $\tilde{s}_t^{(i)} = s_t + \epsilon_t^{(i)}$ denote the perturbed current state, where $\epsilon_t^{(i)} \sim \mathcal{N}(0, \sigma^2 I)$, let a_{t-1} denote the previous action, and let $H_t = \{s_{t-L}, \ldots, s_{t-1}\}$ denote the recent history of length L. The conditional diffusion model produces the denoised state $\hat{s}_t = D_\phi(\tilde{s}_t, a_{t-1}, H_t)$, where $D_\phi(\cdot)$ denotes conditional reverse diffusion with parameters ϕ. In our implementation, the predictor takes $(\tilde{s}_t, a_{t-1}, H_t)$ as input and performs reverse diffusion from a prescribed timestep to a cleaner state representation $\hat{s}_t$. CDS-PPO generates m Gaussian-perturbed copies of the current state and maps them through $D_\phi(\cdot)$. The resulting denoised states are then fed into the PPO policy, which parameterizes a Gaussian action distribution for continuous control: $(\mu_t^{(i)}, \sigma_t^{(i)}) = \pi_\theta(\hat{s}_t^{(i)})$, for $i = 1, \ldots, m$.

The resulting action distributions are aggregated to form the smoothed policy $\tilde{\pi}_\theta(a_t \mid s_t) = \mathcal{N}(\tilde{\mu}_t, \tilde{\Sigma}_t)$. Here $\pi_\theta(\cdot)$ is the PPO policy network with parameters θ, and $\mu_t^{(i)}$ and $\sigma_t^{(i)}$ are the mean and standard deviation predicted from $\hat{s}_t^{(i)}$. For each action dimension $d \in \{1, \ldots, D_a\}$, the smoothed mean and diagonal covariance are defined by the median value Chiang et al. (2020) as

$$\tilde{\mu}_{t,d} = \sup\left\{ u \in \mathbb{R} \;\middle|\; \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\Big[\mu_{t,d}^{(i)} \le u\Big] \le \frac{1}{2} \right\}, \qquad \tilde{\Sigma}_{t,d} = \sup\left\{ v \in \mathbb{R} \;\middle|\; \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\Big[(\sigma_{t,d}^{(i)})^2 \le v\Big] \le \frac{1}{2} \right\}, \tag{10}$$

where $\mu_{t,d}^{(i)}$ and $\sigma_{t,d}^{(i)}$ denote the d-th coordinates of $\mu_t^{(i)}$ and $\sigma_t^{(i)}$, respectively, and $\tilde{\Sigma}_t = \mathrm{diag}(\tilde{\Sigma}_{t,1}, \ldots, \tilde{\Sigma}_{t,D_a})$. Since majority voting, commonly used in randomized smoothing for discrete action spaces, is not directly applicable to continuous action spaces, this aggregation takes the median independently in each action dimension, making it suitable for continuous control. It preserves the central tendency of the action distribution and reduces the influence of outlier predictions induced by perturbations.

Loss Function The optimization of CDS-PPO couples policy learning with conditional diffusion training. At each training iteration, CDS-PPO first collects rollouts using the smoothed policy defined above, and then updates both the conditional diffusion model and the PPO policy using the collected rollouts. For the diffusion update, given a training window of length $L + M$, where the first L states form the conditioning history and the next M states are predicted autoregressively, we first corrupt each target state at a sampled diffusion timestep n:

$$\tilde{s}_{t+k}^{(n)} = \sqrt{\bar{\alpha}_n}\, s_{t+k} + \sqrt{1-\bar{\alpha}_n}\, \epsilon_{t+k}, \qquad \epsilon_{t+k} \sim \mathcal{N}(0, I), \quad k = 0, \ldots, M-1. \tag{11}$$

The conditioning history is then updated recursively by using previously denoised states:

$$\breve{H}_{t+k} = \{\breve{s}_{t+k-L}, \ldots, \breve{s}_{t+k-1}\}, \qquad \breve{s}_j = \begin{cases} s_j, & j < t, \\ D_\phi(\tilde{s}_j^{(n)}, a_{j-1}, \breve{H}_j), & t \le j < t+k. \end{cases} \tag{12}$$

The resulting CDS-PPO diffusion loss is defined as:

$$\mathcal{L}_{\mathrm{D}}^{\mathrm{PPO}} = \mathbb{E}_{n, \{\epsilon_{t+k}\}} \left[\sum_{k=0}^{M-1} (k+1) \left\| \epsilon_\phi\Big(\tilde{s}_{t+k}^{(n)}, a_{t+k-1}, \breve{H}_{t+k}, n\Big) - \epsilon_{t+k} \right\|_2^2 \right]. \tag{13}$$

The factor $(k+1)$ compensates for the increasing difficulty of later autoregressive steps, where the conditioning history contains more model-generated denoised states. This objective exposes the diffusion model to its own denoised states in later conditioning windows, thereby reducing the train-test mismatch caused by recursively using predicted histories. For the PPO update, the policy is optimized using the collected rollouts and the smoothed policy probabilities with the standard clipped surrogate objective,

$$\mathcal{L}_{\mathrm{PPO}} = \mathbb{E}_t \Big[\min\Big(r_t(\theta)\hat{A}_t,\ \mathrm{clip}(r_t(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\hat{A}_t \Big) \Big], \tag{14}$$

where $r_t(\theta) = \tilde{\pi}_\theta(a_t \mid \hat{s}_t) / \tilde{\pi}_{\theta_{\mathrm{old}}}(a_t \mid \hat{s}_t)$ is the PPO importance ratio under the smoothed policy, θ_{old} denotes the previous policy parameters, $\hat{A}_t$ is the advantage estimate computed from the collected rollouts, and ϵ_{clip} is the PPO clipping coefficient. Since the diffusion predictor and PPO are trained jointly, the overall CDS-PPO objective is $\mathcal{L}_{\text{CDS-PPO}} = \lambda_1 \mathcal{L}_{\mathrm{D}}^{\mathrm{PPO}} + \lambda_2 \mathcal{L}_{\mathrm{PPO}}$, where λ_1 and λ_2 balance the diffusion training objective and the PPO optimization objective, respectively. In this way, the diffusion model learns stable state representations under perturbations, while PPO optimizes the policy with respect to the smoothed action distribution induced by those representations.

Reward-based certification. Following global randomized smoothing on the *entire* trajectory, we view the discounted return as a function $G_\pi : \mathbb{R}^{H \times N} \to \mathbb{R}$ that maps a stacked perturbation $\Delta x = [\Delta x_0, \ldots, \Delta x_{H-1}]^\top$ to the cumulative reward, where $G_\pi(\Delta x) \triangleq \sum_{t=0}^{H-1} \gamma^t\, r(s_t, a_t)$ with $a_t = \pi(D(s_t + \Delta x_t))$ and $s_{t+1} \sim P(s_t, a_t)$. Given a smoothing variance σ, draw a global noise vector $\eta = [\eta_0, \ldots, \eta_{H-1}]^\top \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{HN})$ and define the smoothed *p-quantile return* as

$$\mathcal{L}_{\pi,p}(\Delta x) \ \triangleq\ \sup\Big\{ y \in \mathbb{R} \ \Big|\ \mathbb{P}_\eta\big[\, G_\pi(\eta + \Delta x) \le y \,\big] \le p \Big\}.$$

Then, for any adversarial perturbation satisfying $\|\Delta x\|_2 \le B$, we obtain the certified lower bound

$$\mathcal{L}_{\pi,p}(\Delta x) \ \ge\ \mathcal{L}_{\pi,p^\dagger}(\mathbf{0}), \qquad p^\dagger \ =\ \Phi\Big(\Phi^{-1}(p) - \frac{B}{\sigma}\Big), \tag{15}$$

where Φ denotes the CDF of the standard Gaussian distribution and Φ^{-1} is its inverse. Here H is the length of the trajectory and B is the ℓ_2 attack budget; in particular, if each step is bounded by $\|\Delta x_t\|_2 \le \varepsilon$, then $B = \varepsilon\sqrt{H}$. We refer to $\mathcal{L}_{\pi,p^\dagger}(\mathbf{0})$ as the *certified cumulative-reward lower bound*. The proof is provided in Appendix A.3.

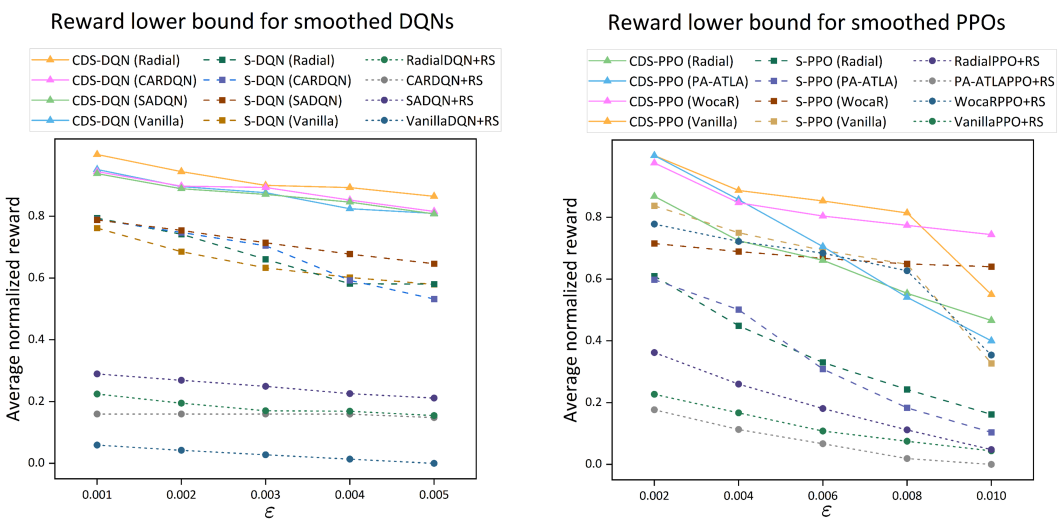


Figure 2: Certified reward lower bound of smoothed agents. Our CDS consistently achieves substantially higher lower bounds than previous smoothed agents.

Action-Bound Certification. For continuous control, certification is cast as a coordinate-wise action bound rather than action invariance. Let $\tilde{\pi}_{\text{det}}(s_t) = \tilde{\mu}_t$ be the deterministic CDS-PPO policy induced by the diffusion-denoised smoothed policy mean. For any $\|\Delta s\|_2 \leq \varepsilon$, its p-quantile action satisfies

$$\tilde{\pi}_{\text{det},\underline{p}}(s_t) \preceq \tilde{\pi}_{\text{det},p}(s_t + \Delta s) \preceq \tilde{\pi}_{\text{det},\bar{p}}(s_t), \qquad \|\Delta s\|_2 \leq \varepsilon, \tag{16}$$

where $\preceq$ is element-wise comparison, $\underline{p} = \Phi(\Phi^{-1}(p) - \varepsilon/\sigma_s)$, and $\bar{p} = \Phi(\Phi^{-1}(p) + \varepsilon/\sigma_s)$. For action coordinate i, this quantile is

$$\tilde{\pi}_{i,\text{det},p}(s_t) = \sup\left\{a_i \in \mathbb{R} \;\middle|\; \mathbb{P}_{\delta\sim\mathcal{N}(0,\sigma_s^2 I)}\left[\mu_{\theta,i}(D_\phi(s_t+\delta, a_{t-1}, H_t)) \leq a_i\right] \leq p\right\}, \tag{17}$$

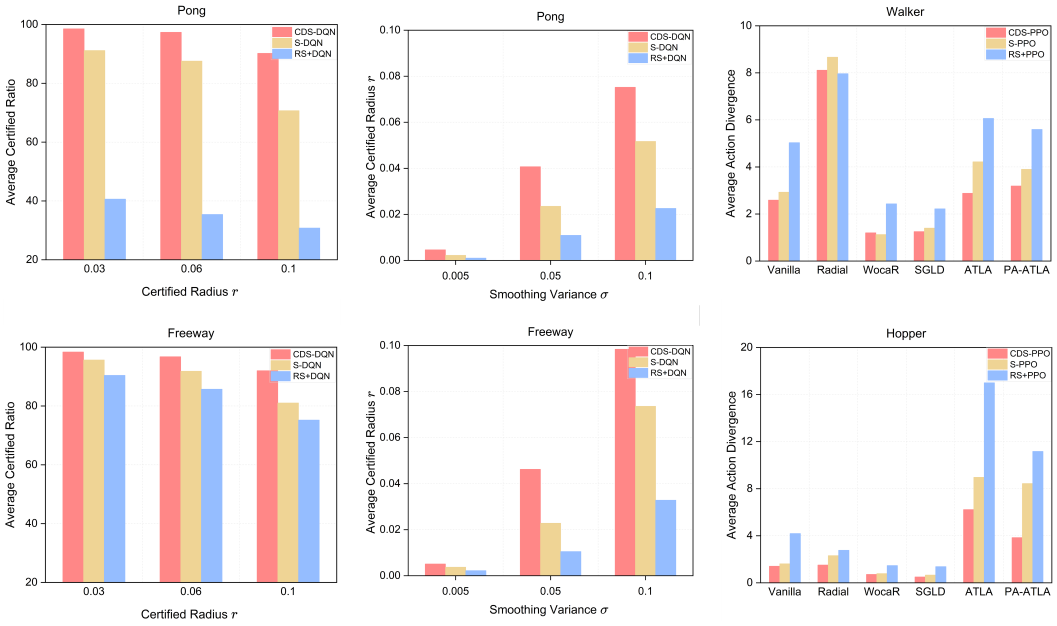
with $\delta \sim \mathcal{N}(0, \sigma_s^2 I)$ and $\mu_{\theta,i}$ the i-th base-policy mean action after denoising by D_ϕ. Thus median smoothing yields certified per-coordinate action intervals; since inter-dimensional dependence is ignored, the certificate is marginal.

5 Evaluation

Setup. We conduct experiments on both Atari Mnih (2013) and MuJoCo Brockman et al. (2016) benchmarks. Specifically, we test CDS-DQN on three Atari environments, namely *Pong*, *Freeway*, and *RoadRunner*, which have high-dimensional pixel inputs and discrete action spaces. Our comparison baselines include non-smoothed robust agents: VanillaDQN, RadialDQN Oikarinen et al. (2021), WocaRDQN Liang et al. (2022), CARDQN Li et al. (2024), and SADQN Zhang et al. (2020); previous smoothed agents: S-DQN Sun et al. (2024) and RS Wu et al. (2022a) (VanillaDQN, RadialDQN, WocaRDQN, CARDQN, SADQN). To compare our RL agents with continuous actions, we test CDS-PPO on two continuous control tasks in the MuJoCo environments, namely *Walker* and *Hopper*, which simulate robotic control and have relatively low-dimensional inputs and a continuous action space. Our comparison baselines include non-smoothed robust agents: VanillaPPO, RadialPPO Oikarinen et al. (2021), WocaRPPO Liang et al. (2022), PA-ATLAPPO Sun et al. (2022), SGLDPPO Zhang et al. (2020), and ATLAPPO Zhang et al. (2021); previous smoothed agents: S-PPO Sun et al. (2024) and RS Wu et al. (2022a) (VanillaPPO, RadialPPO, WocaRPPO, PA-ATLAPPO, SGLDPPO, and ATLAPPO).

5.1 Evaluation of Robustness Guarantees

We evaluate CDS certification through four metrics: action- and value-level certificates for DQN, action divergence for PPO, and certified reward lower bounds for both. Together, they assess robustness under ℓ_p-bounded perturbations.



(a) Certified Ratio (b) Certified radius (c) Action Divergence

Figure 3: Certified robustness results from complementary perspectives. (a) Action-level certified ratio of DQN agents, where higher values indicate that more states remain certifiably action-invariant within the target radius. (b) Value-level certified radius of DQN agents under different smoothing variances. (c) Action Divergence (ADIV) of PPO agents, where lower values indicate tighter certified action bounds and more stable smoothed policies.

Action-level certificates for DQN agents. We fix the smoothing variance σ and report the certified ratio $\eta_r = \frac{1}{N}\sum_{t=1}^{N} \mathbf{1}_{\{r_t \geq r\}}$, the fraction of time steps certifiable at target radius r. This episode-level statistic reflects how often the smoothed policy can provably preserve its selected action. As shown in Figure 3a, CDS-DQNs achieve consistently higher ratios than previous smoothed agents (S-DQN, RS+DQN) across radii and environments. More detailed experimental results and discussion can be found in Appendix D.3.

Value-level certificates for DQN agents. For each smoothing variance σ, we average the value-certified radius over all base DQN agents. Figure 3b shows an approximately linear relationship between certified radius and σ, with CDS-DQNs consistently exceeding S-DQNs and RS+DQNs. The ordering is also consistent with the action-level certificate, supporting the robustness gains across certification criteria. More detailed experimental results and discussion can be found in Appendix D.7.

Action Divergence for PPO agents. Following Sun et al. Sun et al. (2024), we use Action Divergence (ADIV) to measure the width of certified action intervals for PPO; larger ADIV implies looser bounds and lower stability. This metric is therefore a direct proxy for worst-case action sensitivity in continuous control. As shown in Figure 3c, CDS-PPO achieves lower ADIV than smoothed PPO baselines, indicating tighter action bounds under ℓ_2 perturbations. More detailed ADIV results can be found in Appendix D.4.

Reward Lower Bound for DQN and PPO agents We also report certified reward lower bounds, which guarantee that return remains above a threshold under any ℓ_2-bounded perturbation. As shown in Figure 2, CDS-DQNs and CDS-PPOs outperform previous smoothed agents on Atari and MuJoCo. More detailed experimental results and discussion can be found in Appendix D.5. This certificate follows the standard global randomized smoothing setting and covers fixed, non-adaptive trajectory-level perturbations; adaptive perturbations are evaluated empirically but are not certified.

Model	PGD attack				PA-AD
ε (ℓ_∞)	0.01	0.03	0.05	0.07	0.05
CDS-DQN (Radial)	0.958	**0.947**	**0.861**	**0.788**	**0.825**
CDS-DQN (CARDQN)	0.941	0.815	0.778	0.697	0.701
CDS-DQN (SADQN)	**0.985**	0.875	0.758	0.713	0.767
CDS-DQN (WocaR)	0.865	0.803	0.756	0.731	0.658
CDS-DQN (Vanilla)	0.852	0.703	0.595	0.528	0.502
S-DQN (SADQN)	0.864	0.709	0.616	0.568	0.542
S-DQN (WocaR)	0.758	0.721	0.528	0.479	0.457
S-DQN (Vanilla)	0.842	0.385	0.086	0	0
SADQN	0.774	0.006	0.006	0.008	0.007
WocaRDQN	0.528	0.021	0.025	0.025	0.011
VanillaDQN	0.333	0	0	0	0
SADQN+RS	0.191	0.036	0.028	0.026	0.03
WocaRDQN+RS	0.118	0.115	0.115	0.115	0.115
CARDQN+RS	0.121	0.121	0.137	0.12	0.112

Model	Attacks				
	Clean	MAD	Min-RS	Optimal	PA-AD
CDS-PPO (Radial)	0.620	0.709	0.628	0.525	0.486
CDS-PPO (WocaR)	0.861	0.770	0.837	**0.688**	0.746
CDS-PPO (SGLD)	0.808	0.801	0.788	0.405	0.798
CDS-PPO (PA-ATLA)	**1.000**	**0.989**	**0.887**	0.476	**0.916**
CDS-PPO (Vanilla)	0.867	0.871	0.824	0.360	0.766
S-PPO (Radial)	0.569	0.509	0.142	0.137	0.197
S-PPO (WocaR)	0.666	0.649	0.494	0.463	0.473
S-PPO (Vanilla)	0.835	0.719	0.379	0.107	0.145
RadialPPO	0.527	0.500	0.167	0.000	0.033
WocaRPPO	0.791	0.691	0.239	0.352	0.278
VanillaPPO	0.775	0.488	0.042	0.012	0.004
RadialPPO+RS	0.478	0.438	0.069	0.028	0.040
WocaRPPO+RS	0.766	0.706	0.166	0.385	0.228
VanillaPPO+RS	0.677	0.482	0.056	0.014	0.025

Table 1: Average normalized robust rewards under adversarial attacks. The left subtable reports DQN results averaged across *Pong*, *Freeway*, and *RoadRunner*. The right subtable reports PPO results averaged across *Walker* and *Hopper*.

5.2 Evaluation of Robust Reward

The robust rewards of our CDS agents under adversarial attacks are shown in Table 1. The reported rewards are first normalized and then averaged across environments. For DQN agents, we evaluate robustness under ℓ_∞ PGD attacks and PA-AD attacks. For PPO agents, we consider a broad set of adversarial attacks, including the Maximal Action Difference (MAD) attack and the Minimum Robust Sarsa (Min-RS) attack proposed by Zhang et al. (2020). In addition, we evaluate our agents against two stronger adversarial attacks proposed in recent works Zhang et al. (2021); Sun et al. (2022). Specifically, Zhang et al. (2021) introduced the Optimal Attack, which employs an adversarial agent to perturb the observed states. Sun et al. (2022) further proposed the strong PA-AD attack, where an adversarial agent first determines the perturbation direction and then applies FGSM to perturb the states along the specified direction.

Results demonstrate that our CDS-DQN and CDS-PPO agents consistently outperform their counterparts across all robust training algorithms, including previous smoothed agents and strong robust baselines. In particular, compared with previous smoothing-based methods such as S-DQN/PPO and DQN/PPO+RS, CDS achieves substantially higher rewards under adversarial attacks. The gains from previous smoothing methods are often limited, especially when the base agent is not robustly trained, e.g., vanilla agents. This indicates that simply applying randomized smoothing only at evaluation time, or using a smoothed policy that remains unstable within the local perturbation neighborhood, cannot effectively improve the robustness of RL agents. By contrast, our CDS-DQN/PPO (Vanilla) agents already exhibit stronger robustness than robustly trained or smoothed baselines, such as S-DQN (WocaR) and RadialPPO+RS, without further combining CDS with other robust agents. More detailed experimental results and discussion can be found in Appendix D.6, where all test results under attack are reported over six random seeds, with 20 evaluation episodes per seed.

6 Conclusion

In this paper, we present Certified Diffusion Smoothing (CDS), a certifiably robust reinforcement learning framework that integrates randomized smoothing into policy training rather than applying it only as a post-hoc certification wrapper. CDS trains policies over randomized-smoothing neighborhoods and uses the diffusion process to obtain plausible state representations. We instantiated CDS as CDS-DQN and CDS-PPO for discrete and continuous control. Experiments on Atari and MuJoCo demonstrate that CDS achieves stronger certified robustness and higher robust rewards than previous smoothed and robustly trained agents, highlighting CDS as a promising direction for certified reinforcement learning.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing proofs. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, or interpret results. Dataset cleaning and reformatting and qualitative or thematic data analysis were not applicable to this work. We additionally used generative AI tools to search for information and edit the manuscript for readability. We reviewed all AI-assisted work and take responsibility for the final content of this paper, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human participants, personal data, or the release of new datasets. All experiments are conducted in standard simulated reinforcement learning environments. We do not identify any additional ethical concerns beyond those associated with the responsible deployment of reinforcement learning systems.

REPRODUCIBILITY STATEMENT

We provide an anonymous code repository at `https://anonymous.4open.science/r/CDS_DRL`. Detailed training and testing procedures are provided in Appendix C.

REFERENCES

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15453–15461, 2025.

Arpit Bansal, Ping-Yeh Chiang, Michael J Curry, Rajiv Jain, Curtis Wigington, Varun Manjunatha, John P Dickerson, and Tom Goldstein. Certified neural network watermarks with randomized smoothing. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1450–1465. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/bansal22a.html`.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36:41259–41282, 2023.

Vahid Behzadan and Arslan Munir. Whatever does not kill deep reinforcement learning, makes it stronger. *arXiv preprint arXiv:1712.09344*, 2017.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! In *The Eleventh International Conference on Learning Representations*, 2023.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Ping-yeh Chiang, Michael Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection with median smoothing. *Advances in Neural Information Processing Systems*, 33:1275–1286, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Adversarial diffusion for robust reinforcement learning. *arXiv preprint arXiv:2509.23846*, 2025.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Kuba Grudzien, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jifeng Hu, Yanchao Sun, Sili Huang, SiYuan Guo, Hechang Chen, Li Shen, Lichao Sun, Yi Chang, and Dacheng Tao. Instructed diffuser with temporal condition guidance for offline reinforcement learning. *arXiv preprint arXiv:2306.04875*, 2023.

Jongheon Jeong and Jinwoo Shin. Multi-scale diffusion denoised smoothing. *Advances in Neural Information Processing Systems*, 36:67374–67397, 2023.

Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. In *European conference on computer vision*, pp. 274–289. Springer, 2022.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pp. 651–673. PMLR, 2018.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Taisuke Kobayashi. L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4032–4039. IEEE, 2022.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8369–8377, 2023.

Ezgi Korkmaz and Jonah Brown-Cohen. Detecting adversarial directions in deep reinforcement learning to make robust decisions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17534–17543. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/korkmaz23a.html.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*, 2017.

Aounon Kumar, Alexander Levine, and Soheil Feizi. Policy smoothing for provably robust reinforcement learning. *arXiv preprint arXiv:2106.11420*, 2021.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pp. 656–672. IEEE, 2019.

Haoran Li, Zicheng Zhang, Wang Luo, Congying Han, Yudong Hu, Tiande Guo, and Shichen Liao. Towards optimal adversarial robust q-learning with bellman infinity-error. In *ICML*, 2024.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in neural information processing systems*, 35:22547–22561, 2022.

Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: diffusion models as adaptive self-evolving planners. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 20725–20745, 2023.

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

Zeyuan Liu, Zhihe Yang, Jiawei Xu, Rui Yang, Jiafei Lyu, Baoxiang Wang, Yunjian Xu, and Xiu Li. Adg: Ambient diffusion-guided dataset recovery for corruption-robust offline reinforcement learning. *arXiv preprint arXiv:2505.23871*, 2025.

Pierre-François Massiani, Alexander Von Rohr, Lukas Haverbeck, and Sebastian Trimpe. Viability of future actions: Robust reinforcement learning via entropy regularization. In *Seventeenth European Workshop on Reinforcement Learning*, 2024.

Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*, pp. 26087–26105. PMLR, 2023.

Buqing Nie, Jingtian Ji, Yangqing Fu, and Yue Gao. Improve robustness of reinforcement learning against observation perturbations via l^∞ lipschitz policy networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14457–14465, 2024.

Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. Robust deep reinforcement learning through adversarial loss. *Advances in Neural Information Processing Systems*, 34:26156–26167, 2021.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2040–2042, 2018.

Marc Rigter, Jun Yamada, and Ingmar Posner. World models via policy-guided trajectory diffusion. *arXiv preprint arXiv:2312.08533*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pp. 91–100. PMLR, 2022.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in neural information processing systems*, 32, 2019.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33:21945–21957, 2020.

Soumyendu Sarkar, Sajad Mousavi, Ashwin Ramesh Babu, Vineet Gundecha, Sahand Ghorbanpour, and Alexander K Shmakov. Measuring robustness with black-box adversarial attack using reinforcement learning. In *NeurIPS ML Safety Workshop*, 2022.

Yan Scholten, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Hierarchical randomized smoothing. *Advances in Neural Information Processing Systems*, 36:49783–49813, 2023.

Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*, pp. 8707–8718. PMLR, 2020.

Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *Conference on Robot Learning*, pp. 907–917. PMLR, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Chung-En Sun, Sicun Gao, and Tsui-Wei Weng. Breaking the barrier: Enhanced utility and robustness in smoothed drl agents. In *International Conference on Machine Learning*, pp. 46957–46987. PMLR, 2024.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5883–5891, 2020.

Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep RL. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JM2kFbJvvI.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Václav Voráček. Treatment of statistical estimation problems in randomized smoothing for adversarial robustness. *Advances in Neural Information Processing Systems*, 37:133464–133486, 2024.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. Crop: Certifying robust policies for reinforcement learning through functional smoothing. In *International Conference on Learning Representations*, 2022a.

Fan Wu, Linyi Li, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. Copa: Certifying robust policies for offline reinforcement learning against poisoning attacks. In *International Conference on Learning Representations*, 2022b.

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. *Advances in neural information processing systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=sCZbhBvqQaU.

Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning*, pp. 2357–2372. PMLR, 2023.

A PROOFS

A.1 PROOF OF THE ACTION-LEVEL CERTIFICATION

In this section, we provide the formal proof of the certified radius for the proposed *action-level certification*. The proof is based on the proof proposed by Cohen et al. (2019) in Appendix A and Salman et al. (2019) in Appendix A.

Let $\mathcal{S} \subseteq \mathbb{R}^n$ be the state space and $\mathcal{A} = \{1, \ldots, K\}$ be a finite discrete action set. Let $D : \mathbb{R}^n \to \mathbb{R}^n$ denote the reverse diffusion denoising process, and let the base agent induce an *action selection probability* over $\mathcal{A}$ after denoising:

$$q : \mathbb{R}^n \times \mathcal{A} \to [0, 1], \qquad q(x, a) := \Pr\big(\pi(x) = a\big), \tag{18}$$

where $\sum_{a \in \mathcal{A}} q(x, a) = 1$ for any x.

For a smoothing variance $\sigma > 0$, let $\delta \sim \mathcal{N}(0, \sigma^2 I_n)$. Define the *action probability function* (population form) as

$$f(s, a) := \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I_n)} \Big[q\big(D(s + \delta), a\big) \Big]. \tag{19}$$

In practice, we estimate equation 19 by Monte Carlo as in the main text: draw $\delta^{(i)} \sim \mathcal{N}(0, \sigma^2 I_n)$ i.i.d., form the multiset $\mathcal{M}(s) = \{\pi(D(s + \delta^{(i)}))\}_{i=1}^N$, and define

$$\widehat{f}(s, a) = \frac{\mathrm{Count}(\mathcal{M}(s), a)}{N}. \tag{20}$$

Note that $\widehat{f}(s, a)$ is an unbiased estimator of $f(s, a)$.

The *action-smoothed policy* is

$$\tilde{\pi}_a(s) = \arg\max_{a \in \mathcal{A}} f(s, a). \tag{21}$$

Fix a state s_t. Let

$$a_A = \tilde{\pi}_a(s_t), \quad p_A = f(s_t, a_A), \quad p_B = \max_{a \in \mathcal{A} \setminus \{a_A\}} f(s_t, a). \tag{22}$$

Lemma 1 For any action $a \in \mathcal{A}$, the mapping $s \mapsto f(s, a)$ is $\frac{1}{\sigma}\sqrt{\frac{2}{\pi}}$-Lipschitz:

$$\big|f(s_1, a) - f(s_2, a)\big| \le \frac{1}{\sigma}\sqrt{\frac{2}{\pi}}\, \|s_1 - s_2\|_2, \qquad \forall s_1, s_2 \in \mathbb{R}^n. \tag{23}$$

Proof. Fix a. Define the bounded function

$$g_a(t) \triangleq q(D(t), a) \in [0, 1]. \tag{24}$$

Then equation 19 can be rewritten as a Gaussian convolution:

$$f(s, a) = (g_a * \mathcal{N}(0, \sigma^2 I_n))(s) = \frac{1}{(2\pi)^{n/2}\sigma^n} \int_{\mathbb{R}^n} g_a(t) \exp\Big(- \frac{\|s - t\|_2^2}{2\sigma^2} \Big)\, dt. \tag{25}$$

Differentiate under the integral sign:

$$\nabla_s f(s, a) = \frac{1}{(2\pi)^{n/2}\sigma^n} \int_{\mathbb{R}^n} -\frac{1}{\sigma^2}(s - t)\, g_a(t) \exp\Big(- \frac{\|s - t\|_2^2}{2\sigma^2} \Big)\, dt. \tag{26}$$

Let u be any unit vector. Using $0 \le g_a(t) \le 1$,

$$\begin{aligned} u^\top \nabla_s f(s, a) &\le \frac{1}{(2\pi)^{n/2}\sigma^n} \int_{\mathbb{R}^n} \frac{1}{\sigma^2} \big|u^\top (s - t)\big| \exp\Big(- \frac{\|s - t\|_2^2}{2\sigma^2} \Big)\, dt \\ &= \frac{1}{\sigma^2}\, \mathbb{E}_{Z \sim \mathcal{N}(0, \sigma^2 I_n)} \big[|u^\top Z| \big]. \end{aligned} \tag{27}$$

Since $u^\top Z \sim \mathcal{N}(0, \sigma^2)$ in one dimension,

$$\mathbb{E}[|u^\top Z|] = \sigma\sqrt{\frac{2}{\pi}}. \tag{28}$$

Hence $u^\top \nabla_s f(s, a) \le \frac{1}{\sigma}\sqrt{\frac{2}{\pi}}$ for all unit u, implying $\|\nabla_s f(s, a)\|_2 \le \frac{1}{\sigma}\sqrt{\frac{2}{\pi}}$. The Lipschitz bound equation 23 follows from the mean value theorem. □

Lemma 2 For any fixed $a \in \mathcal{A}$, the mapping

$$s \mapsto \sigma\, \Phi^{-1}(f(s,a)) \tag{29}$$

is 1-Lipschitz:

$$\Big|\sigma\Phi^{-1}(f(s_1,a)) - \sigma\Phi^{-1}(f(s_2,a))\Big| \le \|s_1 - s_2\|_2, \qquad \forall s_1, s_2 \in \mathbb{R}^n. \tag{30}$$

Proof. Fix a and abbreviate $f(s) \equiv f(s,a)$. On points where $f(s) \in (0,1)$, Φ^{-1} is differentiable and

$$\nabla_s \Phi^{-1}(f(s)) = \frac{\nabla_s f(s)}{\Phi'\big(\Phi^{-1}(f(s))\big)}, \tag{31}$$

where Φ' is the standard normal density.

It suffices to show $\|\sigma\nabla_s\Phi^{-1}(f(s))\|_2 \le 1$. Equivalently, for any unit direction u,

$$u^\top\big(\sigma\nabla_s\Phi^{-1}(f(s))\big) \le 1. \tag{32}$$

Using equation 31, equation 32 follows from

$$u^\top\big(\sigma\nabla_s f(s)\big) \le \Phi'\big(\Phi^{-1}(f(s))\big). \tag{33}$$

We next express the left-hand side as an expectation over $\delta \sim \mathcal{N}(0, \sigma^2 I_n)$. Starting from equation 19 and differentiating the Gaussian density (as in Lemma 1), one obtains

$$u^\top\big(\sigma\nabla_s f(s)\big) = \frac{1}{\sigma}\,\mathbb{E}_\delta\Big[g_a(s+\delta)\,(\delta^\top u)\Big], \tag{34}$$

where $g_a(t) = q(D(t), a) \in [0,1]$.

Now consider the following constrained maximization problem: among all measurable functions $\psi(\delta) \in [0,1]$ satisfying the mean constraint

$$\mathbb{E}[\psi(\delta)] = f(s), \tag{35}$$

maximize $\mathbb{E}[\psi(\delta)\,(\delta^\top u)]$. Because $\delta^\top u$ is a one-dimensional Gaussian, the maximizer is achieved by placing as much mass as possible on larger values of $\delta^\top u$ under the same mean equation 35, i.e., by a hard threshold rule. Concretely, define the threshold function

$$\psi^\star(\delta) \triangleq \begin{cases} 1, & \delta^\top u \ge \tau, \\ 0, & \delta^\top u < \tau, \end{cases} \tag{36}$$

with τ chosen such that $\mathbb{E}[\psi^\star(\delta)] = f(s)$. Since $\delta^\top u \sim \mathcal{N}(0,\sigma^2)$, this constraint implies

$$\tau = -\sigma\,\Phi^{-1}(f(s)). \tag{37}$$

Why $\psi^\star$ is optimal. Assume some ψ satisfying equation 35 achieves the maximum but differs from $\psi^\star$ on a set of nonzero measure. Let

$$\Omega_+ \triangleq \{\delta : \psi^\star(\delta) > \psi(\delta)\}, \qquad \Omega_- \triangleq \{\delta : \psi^\star(\delta) < \psi(\delta)\}. \tag{38}$$

Because ψ and $\psi^\star$ have the same mean, the total mass deficit on Ω_+ equals the total mass surplus on Ω_-:

$$\int_{\Omega_+} \big(\psi^\star - \psi\big)\, d\mu = \int_{\Omega_-} \big(\psi - \psi^\star\big)\, d\mu, \tag{39}$$

where μ is the Gaussian measure of δ. By construction of $\psi^\star$ as a threshold on $\delta^\top u$, every point in Ω_+ has *larger* projection $\delta^\top u$ than every point in Ω_- (up to a null set). Therefore, moving an infinitesimal amount of mass from Ω_- to Ω_+ preserves the mean constraint but strictly increases $\mathbb{E}[\psi(\delta)(\delta^\top u)]$. This contradicts the maximality of ψ; hence $\psi^\star$ attains the supremum.

Applying this optimality to equation 34 gives

$$u^\top\big(\sigma\nabla_s f(s)\big) \le \frac{1}{\sigma}\,\mathbb{E}_\delta\Big[\psi^\star(\delta)\,(\delta^\top u)\Big]. \tag{40}$$

Let $Z \triangleq \delta^\top u \sim \mathcal{N}(0, \sigma^2)$ and substitute $\tau = -\sigma\Phi^{-1}(f(s))$:

$$\frac{1}{\sigma}\,\mathbb{E}\Big[\psi^\star(\delta)\,(\delta^\top u)\Big] = \frac{1}{\sigma}\,\mathbb{E}\Big[Z \;\text{ subject to } Z \geq -\sigma\Phi^{-1}(f(s))\Big]. \tag{41}$$

A standard truncated-normal identity (via integration by parts) yields

$$\frac{1}{\sigma}\,\mathbb{E}\Big[Z \;\text{ subject to } Z \geq c\Big] = \phi(c/\sigma), \tag{42}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} = \Phi'(x)$. Setting $c = -\sigma\Phi^{-1}(f(s))$ gives

$$\frac{1}{\sigma}\,\mathbb{E}\Big[Z \;\text{ subject to } Z \geq -\sigma\Phi^{-1}(f(s))\Big] = \phi\big(\Phi^{-1}(f(s))\big) = \Phi'\big(\Phi^{-1}(f(s))\big). \tag{43}$$

Combining equation 40 and equation 43 establishes equation 33, hence equation 32. Therefore $\|\sigma\nabla_s\Phi^{-1}(f(s))\|_2 \leq 1$, which implies the 1-Lipschitz claim equation 30. □

Theorem 1 Let s_t be fixed and define p_A, p_B as in equation 22. Define

$$R_t \triangleq \frac{\sigma}{2}\Big(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\Big). \tag{44}$$

Then for any perturbation Δs with $\|\Delta s\|_2 \leq R_t$, the action-smoothed decision is invariant:

$$\tilde{\pi}_a(s_t + \Delta s) = \tilde{\pi}_a(s_t). \tag{45}$$

Equivalently, any Δs that changes the action-smoothed decision must satisfy $\|\Delta s\|_2 \geq R_t$.

Proof. Suppose a perturbation Δs changes the action-smoothed decision away from a_A. Then, at $s_t + \Delta s$, there exists some competing action $a' \neq a_A$ that is at least as probable as a_A, yielding

$$f(s_t + \Delta s, a_A) \leq f(s_t + \Delta s, a'). \tag{46}$$

By Lemma 2 applied to action a_A,

$$\sigma\Phi^{-1}\big(f(s_t, a_A)\big) - \sigma\Phi^{-1}\big(f(s_t + \Delta s, a_A)\big) \leq \|\Delta s\|_2. \tag{47}$$

Using equation 46 and the monotonicity of Φ^{-1},

$$\Phi^{-1}\big(f(s_t + \Delta s, a_A)\big) \leq \Phi^{-1}\big(f(s_t + \Delta s, a')\big), \tag{48}$$

hence equation 47 implies

$$\sigma\Phi^{-1}(p_A) - \sigma\Phi^{-1}\big(f(s_t + \Delta s, a')\big) \leq \|\Delta s\|_2. \tag{49}$$

Similarly, apply Lemma 2 to action a':

$$\sigma\Phi^{-1}\big(f(s_t + \Delta s, a')\big) - \sigma\Phi^{-1}\big(f(s_t, a')\big) \leq \|\Delta s\|_2. \tag{50}$$

Adding equation 49 and equation 50 cancels the middle term and yields

$$\sigma\Phi^{-1}(p_A) - \sigma\Phi^{-1}\big(f(s_t, a')\big) \leq 2\|\Delta s\|_2. \tag{51}$$

Since a_B is the runner-up action at s_t, $f(s_t, a') \leq p_B$, and monotonicity of Φ^{-1} gives $\Phi^{-1}(f(s_t, a')) \leq \Phi^{-1}(p_B)$. Therefore,

$$\|\Delta s\|_2 \geq \frac{\sigma}{2}\Big(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\Big) = R_t. \tag{52}$$

Therefore, any perturbation with $\|\Delta s\|_2 < R_t$ cannot change the action-smoothed decision, proving equation 45. □

Theorem 2 Let $\widehat{f}_{\mathrm{sel}}$ be a Monte Carlo estimator used only to select candidate actions, and let $\widehat{f}_{\mathrm{cert}}$ be an independent Monte Carlo estimator using N_{cert} fresh samples for certification. Let

$$\hat{a}_A = \arg\max_{a\in\mathcal{A}} \widehat{f}_{\mathrm{sel}}(s_t, a), \qquad \hat{a}_B = \arg\max_{a\in\mathcal{A}\setminus\{\hat{a}_A\}} \widehat{f}_{\mathrm{sel}}(s_t, a), \tag{53}$$

and denote $\hat{p}_A = \widehat{f}_{\mathrm{cert}}(s_t, \hat{a}_A)$ and $\hat{p}_B = \widehat{f}_{\mathrm{cert}}(s_t, \hat{a}_B)$. Fix a one-sided confidence parameter $\alpha \in (0, 1)$ and define

$$\Delta = \sqrt{\frac{1}{2N_{\mathrm{cert}}} \ln \frac{1}{\alpha}}. \tag{54}$$

Then, with probability at least $1 - 2\alpha$ over the sampling randomness,

$$f(s_t, \hat{a}_A) \geq \hat{p}_A - \Delta, \qquad f(s_t, \hat{a}_B) \leq \hat{p}_B + \Delta. \tag{55}$$

Consequently, a valid certified radius (with confidence at least $1 - 2\alpha$) is

$$\widehat{R}_t = \frac{\sigma}{2}\Big(\Phi^{-1}(\hat{p}_A - \Delta) - \Phi^{-1}(\hat{p}_B + \Delta)\Big), \tag{56}$$

provided $\hat{p}_A - \Delta > \hat{p}_B + \Delta$ and both arguments of Φ^{-1} lie in $(0, 1)$. Under these conditions, for all Δs with $\|\Delta s\|_2 \leq \widehat{R}_t$, we have $\tilde{\pi}_a(s_t + \Delta s) = \tilde{\pi}_a(s_t)$ with probability at least $1 - 2\alpha$.

Proof. Because the selection and certification samples are independent, we condition on the selected actions $\hat{a}_A, \hat{a}_B$ and treat them as fixed during certification. For any fixed selected action a, the certification count is the sum of N_{cert} i.i.d. trials, each contributing either 0 or 1 to the count, with mean $N_{\mathrm{cert}} f(s_t, a)$. Therefore the empirical frequency $\widehat{f}_{\mathrm{cert}}(s_t, a)$ satisfies a Hoeffding-type concentration bound: for any $t \geq 0$,

$$\Pr\left(\widehat{f}_{\mathrm{cert}}(s_t, a) - f(s_t, a) \geq t\right) \leq \exp(-2N_{\mathrm{cert}} t^2), \tag{57}$$

and similarly,

$$\Pr\left(f(s_t, a) - \widehat{f}_{\mathrm{cert}}(s_t, a) \geq t\right) \leq \exp(-2N_{\mathrm{cert}} t^2). \tag{58}$$

Setting $t = \Delta$ in equation 54 yields the one-sided bounds

$$\Pr\left(f(s_t, a) \geq \widehat{f}_{\mathrm{cert}}(s_t, a) - \Delta\right) \geq 1 - \alpha, \qquad \Pr\left(f(s_t, a) \leq \widehat{f}_{\mathrm{cert}}(s_t, a) + \Delta\right) \geq 1 - \alpha. \tag{59}$$

Apply the lower bound to $a = \hat{a}_A$ and the upper bound to $a = \hat{a}_B$, and use the union bound to obtain equation 55 with probability at least $1 - 2\alpha$.

Condition on the event equation 55. Define

$$\underline{p}_A = \hat{p}_A - \Delta, \qquad \overline{p}_B = \hat{p}_B + \Delta, \tag{60}$$

so that $f(s_t, \hat{a}_A) \geq \underline{p}_A$ and $f(s_t, \hat{a}_B) \leq \overline{p}_B$. Now apply Theorem 1 with (p_A, p_B) replaced by $(\underline{p}_A, \overline{p}_B)$ to conclude that any perturbation changing the smoothed decision must satisfy

$$\|\Delta s\|_2 \geq \frac{\sigma}{2}\Big(\Phi^{-1}(\underline{p}_A) - \Phi^{-1}(\overline{p}_B)\Big) = \widehat{R}_t, \tag{61}$$

which proves the claimed practical certificate. □

A.2 Proof of the value-level certification

In this section, we provide the formal proof of the certified radius for the proposed *value-level certification*. The proof is based on the proof proposed by Salman et al. (2019) and Wu et al. (2022a) in Appendix A.

Proof of value-level certification. Fix time step t and write $s := s_t$. Let $\sigma > 0$ and let Φ be the CDF of $\mathcal{N}(0, 1)$. Assume $Q^{(t)}_{\max} > Q^{(t)}_{\min}$ so that the local normalization in equation 8 is well-defined.

For each action $a \in \mathcal{A}$, using the same local bounds $(Q^{(t)}_{\min}, Q^{(t)}_{\max})$ at time t, define the affine-rescaled map

$$f_a(x) \;:=\; \mathrm{clip}_{[0,1]}\left(\frac{Q^{\pi}(D(x), a) - Q^{(t)}_{\min}}{Q^{(t)}_{\max} - Q^{(t)}_{\min}}\right), \qquad x \in \mathbb{R}^n.$$

Let $\delta \sim \mathcal{N}(0, \sigma^2 I)$. Then the normalized smoothed value in equation 8 can be written as

$$\hat{Q}^\pi(s,a) = \mathbb{E}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}[f_a(s+\delta)] \in [0,1]. \tag{62}$$

The value-level smoothed action is defined by ranking these clipped normalized smoothed values.

We now establish a key smoothness property for the probit transform of the Gaussian-smoothed function. Let $Z \sim \mathcal{N}(0, I)$ and define

$$\bar{f}_a(x) \;:=\; \mathbb{E}[f_a(x+\sigma Z)].$$

By equation 62, $\bar{f}_a(s) = \hat{Q}^\pi(s,a)$. Consider the scalar map

$$\eta_a(x) \;:=\; \sigma\, \Phi^{-1}(\bar{f}_a(x)).$$

We claim that η_a is 1-Lipschitz w.r.t. ℓ_2, i.e.

$$|\eta_a(x) - \eta_a(y)| \le \|x-y\|_2, \qquad \forall x, y \in \mathbb{R}^n. \tag{63}$$

To prove equation 63, it suffices to show that for every unit vector u, the directional derivative satisfies $u^\top \nabla \eta_a(x) \le 1$ for all x.

Since $\bar{f}_a$ is a Gaussian convolution (Weierstrass transform), it is differentiable and we may differentiate under the integral sign:

$$\nabla \bar{f}_a(x) = \nabla \mathbb{E}\big[f_a(x+\sigma Z)\big] = \frac{1}{\sigma}\, \mathbb{E}\big[f_a(x+\sigma Z)\, Z\big]. \tag{64}$$

Let $p := \bar{f}_a(x) \in [0,1]$. Using the chain rule for Φ^{-1} and letting $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$ denote the standard normal density, we have

$$\nabla \eta_a(x) = \sigma \cdot \nabla \Phi^{-1}(p) = \sigma \cdot \frac{\nabla \bar{f}_a(x)}{\phi(\Phi^{-1}(p))}. \tag{65}$$

Combining equation 64 and equation 65 yields, for any unit u,

$$u^\top \nabla \eta_a(x) = \frac{1}{\phi(\Phi^{-1}(p))}\, \mathbb{E}\big[f_a(x+\sigma Z)\,(u^\top Z)\big]. \tag{66}$$

Thus, it remains to upper bound $\mathbb{E}[f_a(x+\sigma Z)\,(u^\top Z)]$ in terms of p. Let $X := u^\top Z \sim \mathcal{N}(0,1)$. Among all measurable functions $g : \mathbb{R}^n \to [0,1]$ that satisfy the mean constraint $\mathbb{E}[g(Z)] = p$, the quantity $\mathbb{E}[g(Z)\,X]$ is maximized by placing as much mass as possible of $g(Z)$ on large values of X. Formally, consider the threshold function

$$h(Z) \;:=\; \mathbf{1}\{u^\top Z \ge \tau\}, \qquad \text{where } \tau \text{ is chosen s.t. } \mathbb{E}[h(Z)] = p.$$

Because $X \sim \mathcal{N}(0,1)$, the condition $\mathbb{P}(X \ge \tau) = p$ implies $\tau = -\Phi^{-1}(p)$. We claim that for any g with $0 \le g \le 1$ and $\mathbb{E}[g(Z)] = p$,

$$\mathbb{E}[g(Z)\,X] \le \mathbb{E}[h(Z)\,X]. \tag{67}$$

To see why equation 67 holds, assume for contradiction that some g attains a larger value. Let $\Omega_+ := \{z : h(z) > g(z)\}$ and $\Omega_- := \{z : h(z) < g(z)\}$. Since $\mathbb{E}[h(Z)] = \mathbb{E}[g(Z)] = p$, we have $\int_{\Omega_+} (h-g)\, d\mu = \int_{\Omega_-} (g-h)\, d\mu$, where μ is the Gaussian measure. Define $\tilde{g} := g + (h-g)\mathbf{1}_{\Omega_+} - (g-h)\mathbf{1}_{\Omega_-}$. Then $\tilde{g} \in [0,1]$ and $\mathbb{E}[\tilde{g}(Z)] = p$. Moreover, by the definition of h as a threshold on $X = u^\top z$, we have $X(z) > X(z')$ for all $z \in \Omega_+$ and $z' \in \Omega_-$. Therefore, moving mass from Ω_- to Ω_+ (which is exactly what passing from g to $\tilde{g}$ does) strictly increases $\int g(z) X(z)\, d\mu(z)$, contradicting the assumed optimality of g. Hence h maximizes the objective and equation 67 holds.

It follows that

$$\begin{aligned} \mathbb{E}[f_a(x+\sigma Z)\,(u^\top Z)] &\le \mathbb{E}[X\, \mathbf{1}\{X \ge -\Phi^{-1}(p)\}] && (68)\\ &= \int_{-\Phi^{-1}(p)}^{\infty} x\phi(x)\, dx = \phi(\Phi^{-1}(p)). && (69) \end{aligned}$$

Substituting equation 68–equation 69 into equation 66 yields $u^\top \nabla \eta_a(x) \le 1$ for every unit u, hence $\|\nabla \eta_a(x)\|_2 \le 1$ for all x, which implies equation 63. In particular,

$$|\eta_a(s) - \eta_a(s+\Delta)| \le \|\Delta\|_2, \qquad \forall \Delta \in \mathbb{R}^n. \tag{70}$$

Now let a_A be a top action at s under $\hat{Q}^\pi$, and a_B be the runner-up action:

$$a_A = \arg\max_{a \in \mathcal{A}} \hat{Q}^\pi(s,a), \qquad a_B = \arg\max_{a \in \mathcal{A}\setminus\{a_A\}} \hat{Q}^\pi(s,a).$$

Consider any perturbation Δ. From equation 70 applied to a_A and a_B, we have

$$\eta_{a_A}(s) - \eta_{a_A}(s+\Delta) \le \|\Delta\|_2, \tag{71}$$
$$\eta_{a_B}(s+\Delta) - \eta_{a_B}(s) \le \|\Delta\|_2. \tag{72}$$

Suppose that the action selected by the locally smoothed policy changes under Δ. Then there exists some competing action $a' \neq a_A$ that is at least as good as the original top action at the perturbed input:

$$\hat{Q}^\pi(s+\Delta, a_A) \le \hat{Q}^\pi(s+\Delta, a').$$

Since Φ^{-1} is strictly increasing and $\eta_a = \sigma\Phi^{-1}(\hat{Q}^\pi(\cdot,a))$, the above implies

$$\eta_{a_A}(s+\Delta) \le \eta_{a'}(s+\Delta). \tag{73}$$

Replacing a_B by a' in equation 72 and adding equation 71, equation 72, and equation 73 gives

$$\eta_{a_A}(s) - \eta_{a'}(s) \le 2\|\Delta\|_2.$$

Since a_B is the runner-up action at s, $\hat{Q}^\pi(s,a') \le \hat{Q}^\pi(s,a_B)$, and therefore $\eta_{a'}(s) \le \eta_{a_B}(s)$. Thus, any perturbation that can change the selected action must satisfy

$$\|\Delta\|_2 \ge \frac{1}{2}\big(\eta_{a_A}(s) - \eta_{a_B}(s)\big) = \frac{\sigma}{2}\Big(\Phi^{-1}(\hat{Q}^\pi(s,a_A)) - \Phi^{-1}(\hat{Q}^\pi(s,a_B))\Big). \tag{74}$$

Consequently, for every perturbation obeying

$$\|\Delta\|_2 < R_t \quad \text{with} \quad R_t := \frac{\sigma}{2}\Big(\Phi^{-1}(\hat{Q}^\pi(s,a_A)) - \Phi^{-1}(\hat{Q}^\pi(s,a_B))\Big),$$

the inequality equation 74 cannot hold, which means that no action flip is possible. Equivalently, the value-level selected action remains a_A for all $\|\Delta\|_2 < R_t$, proving equation 9.

Finally, the runner-up action a_B (i.e., the maximizer of $\hat{Q}^\pi(s,a)$ among $a \neq a_A$) minimizes the lower bound in equation 74 over all competing actions, hence it is the tightest certified radius obtainable from this argument at state s_t. □

A.3 PROOF FOR THE REWARD-BASED CERTIFICATION

In this section, we provide the formal proof of the reward lower bound introduced in Section 4.2. The proof is based on the proof proposed by Chiang et al. (2020).

Theorem A.1 (Certified lower bound of globally smoothed p-quantile return)**.** *Let $G_\pi : \mathbb{R}^{HN} \to \mathbb{R}$ map a stacked trajectory perturbation $\Delta x = [\Delta x_0, \dots, \Delta x_{H-1}]^\top$ to the discounted return $G_\pi(\Delta x) = \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)$, where $a_t = \pi(s_t + \Delta x_t)$ and $s_{t+1} \sim P(s_t, a_t)$. For stochastic environments, G_π denotes the return after marginalizing over environment randomness, or equivalently the return conditional on fixed environment randomness. Let $\eta \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{HN})$ and define the smoothed p-quantile return*

$$\mathcal{L}_{\pi,p}(\Delta x) = \sup \Big\{ y \in \mathbb{R} \;\Big|\; \mathbb{P}_\eta\big[G_\pi(\eta + \Delta x) \le y\big] \le p \Big\}.$$

Then for any fixed non-adaptive perturbation $\|\Delta x\|_2 \le B$,

$$\mathcal{L}_{\pi,p}(\Delta x) \;\ge\; \mathcal{L}_{\pi,p^\dagger}(\mathbf{0}), \qquad p^\dagger = \Phi\Big(\Phi^{-1}(p) - \frac{B}{\sigma}\Big),$$

where Φ is the CDF of $\mathcal{N}(0,1)$.

Proof. Fix $p \in (0,1)$ and define

$$p^\dagger = \Phi\Big(\Phi^{-1}(p) - \frac{B}{\sigma}\Big), \qquad y_0 := \mathcal{L}_{\pi,p^\dagger}(\mathbf{0}).$$

By the definition of $\mathcal{L}_{\pi,p^\dagger}(\mathbf{0})$, we have

$$\mathbb{P}_\eta\big[G_\pi(\eta) \le y_0\big] \le p^\dagger. \tag{75}$$

Now define the auxiliary function $E : \mathbb{R}^{HN} \to [0,1]$ by

$$E(\Delta x) := \mathbb{E}_{\eta \sim \mathcal{N}(0,\sigma^2 I)}\Big[\mathbf{1}\big\{G_\pi(\eta + \Delta x) \le y_0\big\}\Big] = \mathbb{P}_\eta\big[G_\pi(\eta + \Delta x) \le y_0\big].$$

Consider the function $f(z) := \mathbf{1}\{G_\pi(z) \le y_0\} \in [0,1]$. Then $E(\Delta x) = \hat{f}(\Delta x)$ is exactly the Gaussian smoothing of f. By Lemma 2 of Salman et al. (the nonlinear smoothness property), the mapping $\Delta x \mapsto \sigma\Phi^{-1}(E(\Delta x))$ is 1-Lipschitz, hence for any Δx,

$$\sigma\Phi^{-1}(E(\Delta x)) - \sigma\Phi^{-1}(E(\mathbf{0})) \;\le\; \|\Delta x\|_2. \tag{76}$$

Rearranging equation 76 and using $\|\Delta x\|_2 \le B$ yields

$$\Phi^{-1}(E(\Delta x)) \le \Phi^{-1}(E(\mathbf{0})) + \frac{\|\Delta x\|_2}{\sigma} \le \Phi^{-1}(E(\mathbf{0})) + \frac{B}{\sigma}.$$

By equation 75, $E(\mathbf{0}) = \mathbb{P}_\eta[G_\pi(\eta) \le y_0] \le p^\dagger$, so monotonicity of Φ^{-1} gives

$$\Phi^{-1}(E(\mathbf{0})) \le \Phi^{-1}(p^\dagger) = \Phi^{-1}(p) - \frac{B}{\sigma}.$$

Combining the two displays, we obtain

$$\Phi^{-1}(E(\Delta x)) \le \Big(\Phi^{-1}(p) - \frac{B}{\sigma}\Big) + \frac{B}{\sigma} = \Phi^{-1}(p).$$

Finally, applying the monotonicity of Φ yields

$$E(\Delta x) = \mathbb{P}_\eta\big[G_\pi(\eta + \Delta x) \le y_0\big] \le p.$$

Therefore, y_0 is a feasible candidate in the defining set of $\mathcal{L}_{\pi,p}(\Delta x)$, so

$$\mathcal{L}_{\pi,p}(\Delta x) = \sup \Big\{ y : \mathbb{P}_\eta[G_\pi(\eta + \Delta x) \le y] \le p \Big\} \ge y_0 = \mathcal{L}_{\pi,p^\dagger}(\mathbf{0}).$$

This completes the proof. □

B ALGORITHMS

This section presents the training and certification procedures of Certified Diffusion Smoothing (CDS).

B.1 TRAINING PROCEDURE FOR CDS-DQN

Algorithm 1 summarizes the training procedure for CDS-DQN. At each interaction step, CDS-DQN realizes the randomized-smoothing perturbation through the matched diffusion forward process, denoises the perturbed observations by reverse diffusion, and selects the action through the diffusion-smoothed policy. In our training implementation, we use $m = 10$ perturbed samples at each state. The resulting transition is stored in the replay buffer, after which the Q-network and diffusion denoiser are updated using the TD loss and diffusion loss, respectively.

Algorithm 1 Training procedure for CDS-DQN

Input: Environment $\mathcal{E}$, pretrained Q-network Q_θ, target network Q_{θ^-}, reverse diffusion model D_ϕ, smoothing variance σ, sample size m, replay buffer $\mathcal{B}$.
Output: Trained CDS-DQN parameters (θ, ϕ).
Initialize Q_θ, Q_{θ^-}, D_ϕ, and replay buffer $\mathcal{B}$.
for each environment step **do**
  Observe the current state s_t.
  Find nearest diffusion timestep $t^\star$ such that $\sigma^2 \approx (1 - \bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$.
  for $i = 1$ **to** m **do**
    Draw $\epsilon^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ and generate $\tilde{s}_t^{(i)} = \sqrt{\bar{\alpha}_{t^\star}}\, s_t + \sqrt{1 - \bar{\alpha}_{t^\star}}\, \epsilon^{(i)}$.
    Denoise from timestep $t^\star$: $\hat{s}_t^{(i)} = D_\phi(\tilde{s}_t^{(i)}, t^\star)$.
    Compute the base action prediction $a^{(i)} = \arg\max_a Q_\theta(\hat{s}_t^{(i)}, a)$.
  end for
  Select the smoothed action $a_t = \mathrm{MajorityVote}(\{a^{(i)}\}_{i=1}^m)$ and execute a_t in $\mathcal{E}$.
  Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$.
  Sample a minibatch from $\mathcal{B}$ and update Q_θ using the CDS-DQN TD loss $\mathcal{L}_{\mathrm{Q}}$.
  Update D_ϕ using the diffusion denoising objective $\mathcal{L}_{\mathrm{D}}$.
  Periodically update the target network Q_{θ^-}.
end for
Return (θ, ϕ).

B.2 TRAINING PROCEDURE FOR CDS-PPO

Algorithm 2 summarizes the training procedure for CDS-PPO. CDS-PPO uses a conditional reverse diffusion model that takes the perturbed current state, previous action, and recent history as input. In our training implementation, we use $m = 50$ perturbed samples at each state. The denoised samples are evaluated by a Gaussian PPO policy, and the policy and diffusion model are jointly optimized with the PPO objective and the conditional diffusion loss.

B.3 PROCEDURE FOR ACTION-LEVEL CERTIFICATION

Algorithm 3 gives the procedure for action-level certification. For a state s_t, CDS first aligns the smoothing variance with the diffusion process as in Section 4.1: it finds the nearest timestep $t^\star$ satisfying $\sigma^2 \approx (1 - \bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$ and uses the corresponding forward diffusion step to generate noisy states. To avoid post-selection bias in finite-sample certification, we use one sample set to select the top and runner-up actions and a disjoint sample set to estimate their probabilities. Each noisy state is denoised through the reverse diffusion process $D(\cdot)$, and the action selected by the base policy is recorded. Let a_A be the selected top action and a_B be the selected runner-up action. Following the action-level certificate in Section 4.1, certification succeeds when the lower confidence bound of $f(s_t, a_A)$ remains larger than the upper confidence bound of $f(s_t, a_B)$; the certified ℓ_2 radius is then computed from the corresponding Gaussian quantile gap.

Algorithm 2 Training procedure for CDS-PPO

Input: Environment $\mathcal{E}$, PPO policy π_θ, value function V_ψ, conditional reverse diffusion model D_ϕ, smoothing variance σ, sample size m, rollout horizon T.
Output: Trained CDS-PPO parameters (θ, ψ, ϕ).
Initialize π_θ, V_ψ, and D_ϕ.
for each PPO iteration **do**
  Initialize an empty rollout buffer $\mathcal{R}$.
  for $t = 1$ **to** T **do**
    Observe s_t, previous action a_{t-1}, and history H_t.
    Find nearest diffusion timestep $t^\star$ such that $\sigma^2 \approx (1 - \bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$.
    for $i = 1$ **to** m **do**
      Draw $\epsilon^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ and generate $\tilde{s}_t^{(i)} = \sqrt{\bar{\alpha}_{t^\star}} s_t + \sqrt{1 - \bar{\alpha}_{t^\star}} \epsilon^{(i)}$.
      Denoise with temporal context: $\hat{s}_t^{(i)} = D_\phi(\tilde{s}_t^{(i)}, a_{t-1}, H_t, t^\star)$.
      Evaluate the Gaussian policy $(\mu_t^{(i)}, \Sigma_t^{(i)}) = \pi_\theta(\hat{s}_t^{(i)})$.
    end for
    Form the smoothed policy $\tilde{\pi}_\theta(\cdot|s_t)$ by aggregating the sampled Gaussian policies.
    Sample and execute $a_t \sim \tilde{\pi}_\theta(\cdot|s_t)$, then store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$.
  end for
  Estimate advantages from $\mathcal{R}$.
  Update (θ, ψ) using the PPO clipped objective $\mathcal{L}_{\text{PPO}}$.
  Update D_ϕ using the conditional diffusion loss $\mathcal{L}_{\text{D}}^{\text{PPO}}$.
end for
Return (θ, ψ, ϕ).

Algorithm 3 Procedure for action-level certification

Input: Smoothing variance σ, reverse diffusion process $D(\cdot)$, base policy π, selection sample size M_{sel}, certification sample size M_{cert}, one-sided confidence parameter α.
Output: Selected action a^* and certified radius r for action-level certification.
Initialize empty multisets $\mathcal{M}_{\text{sel}}$ and $\mathcal{M}_{\text{cert}}$ to collect action predictions.
Get the current state s_t from the environment.
Find nearest diffusion timestep $t^\star$ such that $\sigma^2 \approx (1 - \bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$.
for $i = 1$ **to** M_{sel} **do**
  Draw $\epsilon^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ and generate the aligned noisy state by forward diffusion: $\tilde{s}_i = \sqrt{\bar{\alpha}_{t^\star}} s_t + \sqrt{1 - \bar{\alpha}_{t^\star}} \epsilon^{(i)}$.
  Denoise from timestep $t^\star$ via reverse diffusion: $\hat{s}_i = D(\tilde{s}_i, t^\star)$.
  Query the base policy: $a_i = \pi(\hat{s}_i)$.
  Record a_i into the multiset $\mathcal{M}_{\text{sel}}$.
end for
Select $a^* = \arg\max_a \text{Count}(\mathcal{M}_{\text{sel}}, a)$ and $a_B = \arg\max_{a \neq a^*} \text{Count}(\mathcal{M}_{\text{sel}}, a)$.
for $i = 1$ **to** M_{cert} **do**
  Draw fresh $\epsilon^{(i)} \sim \mathcal{N}(0, \mathbf{I})$, generate $\tilde{s}_i = \sqrt{\bar{\alpha}_{t^\star}} s_t + \sqrt{1 - \bar{\alpha}_{t^\star}} \epsilon^{(i)}$, denoise $\hat{s}_i = D(\tilde{s}_i, t^\star)$, and record $\pi(\hat{s}_i)$ into $\mathcal{M}_{\text{cert}}$.
end for
Estimate $p_A = \frac{\text{Count}(\mathcal{M}_{\text{cert}}, a^*)}{M_{\text{cert}}}$ and $p_B = \frac{\text{Count}(\mathcal{M}_{\text{cert}}, a_B)}{M_{\text{cert}}}$.
Calculate confidence interval $\Delta = \sqrt{\frac{1}{2M_{\text{cert}}} \ln \frac{1}{\alpha}}$
if $p_A \geq p_B + 2\Delta$ **then**
  Calculate certified radius $r = \frac{\sigma}{2}(\Phi^{-1}(p_A - \Delta) - \Phi^{-1}(p_B + \Delta))$.
else
  Fail certification, undefined r
end if
Execute the robust action a^* in the environment.
Receive reward r_t and move to the next state s_{t+1}.
Return a^* and r.

B.4 PROCEDURE FOR VALUE-LEVEL CERTIFICATION

Algorithm 4 gives the procedure for value-level certification. Following Section 4.1, CDS certifies robustness by smoothing action values rather than only counting action predictions. For each state s_t, we first align the smoothing variance with the diffusion process as in Section 4.1, find the nearest timestep $t^\star$ satisfying $\sigma^2 \approx (1-\bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$, and generate noisy states using the corresponding forward diffusion step. After reverse diffusion denoising, we normalize and clip each sampled Q-value using independently estimated local bounds $Q^{(t)}_{\max}$ and $Q^{(t)}_{\min}$, and then average the clipped normalized values. Value-level certification selects the action with the largest normalized smoothed value; the certified radius is then obtained from the Gaussian quantile gap between the top action and the runner-up action.

Algorithm 4 Procedure for value-level certification

Input: Smoothing variance σ, reverse diffusion process $D(\cdot)$, action-value function Q^π, sample size M, one-sided confidence parameter α.
Output: Selected action a^* and certified radius r for value-level certification.
Get the current state s_t from the environment.
Find nearest diffusion timestep $t^\star$ such that $\sigma^2 \approx (1-\bar{\alpha}_{t^\star})/\bar{\alpha}_{t^\star}$.
for $j = 1$ **to** M **do**
    Draw $\epsilon^{(j)} \sim \mathcal{N}(0, \mathbf{I})$ and generate the aligned noisy state by forward diffusion: $\tilde{s}_t^{(j)} = \sqrt{\bar{\alpha}_{t^\star}} s_t + \sqrt{1-\bar{\alpha}_{t^\star}}\epsilon^{(j)}$.
    Denoise from timestep $t^\star$ via reverse diffusion: $\hat{s}_t^{(j)} = D(\tilde{s}_t^{(j)}, t^\star)$.
    Compute and store Q-values for all actions: $Q^\pi(\hat{s}_t^{(j)}, a)$.
end for
Estimate local bounds $Q^{(t)}_{\max}$ and $Q^{(t)}_{\min}$ from an independent sample set.
Normalize and clip each sample, then average: $\hat{Q}^\pi(s_t, a) = \frac{1}{M}\sum_{j=1}^{M} \mathrm{clip}_{[0,1]}\left(\frac{Q^\pi\left(\hat{s}_t^{(j)}, a\right) - Q^{(t)}_{\min}}{Q^{(t)}_{\max} - Q^{(t)}_{\min}}\right)$.
Determine the value-level action $a^* = \tilde{\pi}_v(s_t) = \arg\max_{a\in\mathcal{A}} \hat{Q}^\pi(s_t, a)$.
Rank actions to find top-1 $a_A = \arg\max_{a\in\mathcal{A}} \hat{Q}^\pi(s_t, a)$ and runner-up $a_B = \arg\max_{a\in\mathcal{A}\setminus\{a_A\}} \hat{Q}^\pi(s_t, a)$.
Calculate confidence interval $\Delta = \sqrt{\frac{1}{2M}\ln\frac{1}{\alpha}}$.
if $\hat{Q}^\pi(s_t, a_A) \geq \hat{Q}^\pi(s_t, a_B) + 2\Delta$ **then**
    Calculate certified radius $r = \frac{\sigma}{2}\left(\Phi^{-1}(\hat{Q}^\pi(s_t, a_A) - \Delta) - \Phi^{-1}(\hat{Q}^\pi(s_t, a_B) + \Delta)\right)$.
else
    Fail certification, undefined r.
end if
Execute the robust action a^* in the environment.
Receive reward r_t and move to the next state s_{t+1}.

C DETAILED SETTINGS FOR DQN AND PPO

All experiments are conducted on a workstation equipped with four NVIDIA RTX 4090 GPUs, one NVIDIA A100 GPU, 2 TB of storage, and 524 GB of system memory.

C.1 SETTINGS AND TRAINING TIME FOR DQN

Our DQN implementation is based on SADQN Zhang et al. (2020), RadialDQN Oikarinen et al. (2021), WocaRDQN Liang et al. (2022), and CARDQN Li et al. (2024). For Atari experiments, we train CDS-DQN for 500,000 frames on *Pong*, *Freeway*, and *RoadRunner*. On our hardware, training CDS-DQN takes approximately 15 hours, compared with 12 hours for S-DQN and 27, 15, 18, and 14 hours for robustly trained SADQN, RadialDQN, WocaRDQN, and CARDQN agents, respectively.

During training, CDS-DQN is initialized with the corresponding base model. The smoothing variance σ is set to 0.1 for *Pong* and *Freeway*, and 0.05 for *RoadRunner*. All reported attack results are averaged over five episodes, and the reported error bars denote standard deviations across episodes. Test-time runtime comparisons are provided in Table 19.

C.2 Settings and Training Time for PPO

Our PPO implementation is based on WocaRPPO Liang et al. (2022), RadialPPO Oikarinen et al. (2021), ATLAPPO Zhang et al. (2021), PA-ATLAPPO Sun et al. (2022), and SGLDPPO Zhang et al. (2020). CDS-PPO converges within 500,000 frames, taking approximately 12 hours. SA-PPO is trained for 4.0M steps and takes 6.0 hours; ATLA-PPO is trained for 5.0M steps and takes 5.6 hours; PA-ATLA-PPO is trained for 5.0M steps and takes 5.2 hours; RadialPPO is trained for 4.0M steps and takes 3.2 hours; WocaRPPO is trained for 4.0M steps and takes 5 hours; S-PPO trains 15 candidate models, each of which takes approximately 6 hours to train. The smoothing variance σ of CDS-PPO is set to 0.2 in all environments. For all PPO attacks, including MAD, Min-RS, Optimal Attack, and PA-AD attack, the ℓ_∞ attack budget is set to 0.075. All reported attack results are averaged over 50 episodes, with standard deviations reported where applicable. Test-time runtime comparisons are provided in Table 19.

C.3 Limitations

While CDS demonstrates consistent gains in certified robustness and robust rewards, several directions remain open for further study. Our experiments focus on standard Atari and MuJoCo benchmarks, and future work may explore broader environments, richer observation modalities, and more diverse policy architectures. Further improving computational efficiency is also important, as diffusion-based smoothing and certification still introduce relatively high overhead.

D Experiment results

D.1 Controlled ablation of CDS-DQN

To disentangle the contributions of diffusion denoising and Q-network fine-tuning, we evaluate four controlled configurations under identical settings. S-DQN (Vanilla) combines a DnCNN denoiser with a frozen Q-network; Frozen Q + Diffusion replaces DnCNN with the diffusion denoiser while keeping the Q-network frozen; CDS w/o Diffusion removes the denoiser while fine-tuning the Q-network with the randomized-smoothing objective; and CDS-DQN combines diffusion denoising with Q-network fine-tuning. Thus, S-DQN versus Frozen Q + Diffusion isolates the denoiser, whereas Frozen Q + Diffusion versus CDS-DQN isolates Q-network fine-tuning. The CDS w/o Diffusion configuration tests whether the fine-tuned Q-network alone can account for the observed gains.

Table 2: Controlled ablation under ℓ_∞ PGD attacks. Values are average rewards $\pm$ standard deviation over five episodes; higher is better.

Environment	Model	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.03$	$\varepsilon = 0.04$
Pong	VanillaDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
	S-DQN (Vanilla)	**21.0±0.0**	20.5±0.4	**20.2±0.8**	17.6±1.8
	Frozen Q + Diffusion	**21.0±0.0**	20.4±0.4	20.0±0.2	19.0±2.5
	CDS w/o Diffusion	19.0±3.8	-17.6 ± 4.2	-20.1 ± 1.6	-21.0 ± 0.0
	CDS-DQN (Vanilla)	**21.0±0.0**	**20.8±0.2**	**20.2±0.1**	**21.0±0.1**
Freeway	VanillaDQN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	S-DQN (Vanilla)	**32.0±0.4**	30.0±1.0	20.4±1.0	0.0±0.0
	Frozen Q + Diffusion	**32.0±0.6**	31.4±0.2	29.8±0.2	29.3±0.6
	CDS w/o Diffusion	26.5±1.2	14.4±0.6	0.0±0.0	0.0±0.0
	CDS-DQN (Vanilla)	**32.0±0.0**	**32.2±0.0**	**32.4±0.2**	**32.2±0.1**
RoadRunner	VanillaDQN	0 ± 0	0 ± 0	0 ± 0	0 ± 0
	S-DQN (Vanilla)	28750 ± 3784	16480 ± 3416	460 ± 382	0 ± 0
	Frozen Q + Diffusion	28925 ± 4087	20056 ± 3862	10214 ± 4720	7398 ± 3194
	CDS w/o Diffusion	1180 ± 0	64 ± 0	0 ± 0	0 ± 0
	CDS-DQN (Vanilla)	**29320±3841**	**21360±5963**	**15140±3373**	**9500±2913**

The results in Table 2 show that diffusion denoising makes a distinct contribution. With the Q-network frozen, replacing DnCNN by diffusion improves robustness most clearly at larger attack budgets: on Freeway, the reward at $\varepsilon = 0.04$ increases from 0.0 to 29.3, while on RoadRunner it increases from 0 to 7398. The benefit is also visible on Pong, where Frozen Q + Diffusion retains 19.0 reward at $\varepsilon = 0.04$ compared with 17.6 for S-DQN.

Fine-tuning the Q-network provides a complementary improvement. Frozen Q + Diffusion and CDS-DQN use the same denoiser, yet CDS-DQN achieves higher rewards at stronger attacks on Freeway and RoadRunner (32.2 versus 29.3, and 9500 versus 7398, respectively, at $\varepsilon = 0.04$), and preserves the highest reward on Pong. Finally, CDS w/o Diffusion rapidly collapses as the attack strengthens despite using the fine-tuned Q-network. This confirms that the gains of CDS cannot be attributed to Q-network fine-tuning alone: diffusion denoising supplies the principal robustness improvement, with fine-tuning yielding additional gains.

D.2 Sensitivity to the smoothing level σ

We study the sensitivity of CDS to the smoothing level by varying σ independently of the test-time attack budget. For CDS-DQN, we evaluate $\sigma \in \{0.01, 0.05, 0.10, 0.15\}$ under ℓ_∞ PGD attacks with $\varepsilon \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$. For CDS-PPO, we evaluate $\sigma \in \{0.10, 0.20, 0.30\}$ under MAD, Min-RS, Optimal, and PA-AD attacks with $\varepsilon = 0.075$. Results are reported as mean $\pm$ standard deviation.

Table 3: Sensitivity of CDS-DQN to σ under ℓ_∞ PGD attacks.

Agent	Environment	σ	$\varepsilon = .01$	$\varepsilon = .02$	$\varepsilon = .03$	$\varepsilon = .04$	$\varepsilon = .05$
CDS-DQN	*Pong*	0.01	$\mathbf{21.0 \pm 0.0}$	18.2 ± 3.1	13.7 ± 2.5	5.2 ± 2.5	-9.0 ± 4.0
		0.05	$\mathbf{21.0 \pm 0.0}$	$\mathbf{21.0 \pm 0.0}$	$\mathbf{20.5 \pm 0.1}$	17.3 ± 0.6	14.9 ± 1.6
		0.10	$\mathbf{21.0 \pm 0.0}$	20.8 ± 0.2	20.2 ± 0.1	$\mathbf{21.0 \pm 0.1}$	$\mathbf{20.4 \pm 0.1}$
		0.15	20.3 ± 0.7	20.0 ± 0.5	19.3 ± 0.8	18.0 ± 0.8	17.2 ± 2.1
CDS-DQN	*Freeway*	0.01	32.0 ± 0.2	22.3 ± 2.2	19.4 ± 3.0	13.7 ± 4.8	9.0 ± 2.4
		0.05	$\mathbf{32.2 \pm 0.2}$	$\mathbf{32.6 \pm 0.2}$	28.5 ± 0.4	25.6 ± 0.8	20.2 ± 3.0
		0.10	32.0 ± 0.0	32.2 ± 0.0	$\mathbf{32.4 \pm 0.2}$	$\mathbf{32.2 \pm 0.1}$	$\mathbf{32.0 \pm 0.2}$
		0.15	31.5 ± 1.4	31.4 ± 0.9	32.2 ± 1.3	31.0 ± 0.8	31.2 ± 1.4
CDS-DQN	*RoadRunner*	0.01	13255 ± 2631	8903 ± 2572	4290 ± 2374	1379 ± 250	682 ± 33
		0.05	$\mathbf{29320 \pm 3841}$	$\mathbf{21360 \pm 5963}$	15140 ± 3373	9500 ± 2912	4100 ± 3381
		0.10	22036 ± 2539	20874 ± 3268	17235 ± 4950	10748 ± 3075	7253 ± 1839
		0.15	19563 ± 2275	18917 ± 1004	$\mathbf{17626 \pm 1372}$	$\mathbf{12931 \pm 2495}$	$\mathbf{9740 \pm 2072}$

Table 4: Sensitivity of CDS-PPO to σ under attacks with $\varepsilon = 0.075$.

Agent	Environment	σ	Clean	MAD	Min-RS	Optimal	PA-AD
CDS-PPO	*Walker*	0.10	4178 ± 130	4557 ± 176	3966 ± 358	$\mathbf{2460 \pm 95}$	2785 ± 390
		0.20	$\mathbf{4752 \pm 54}$	$\mathbf{4709 \pm 124}$	$\mathbf{4596 \pm 307}$	2368 ± 353	$\mathbf{4306 \pm 422}$
		0.30	4473 ± 60	4292 ± 241	3580 ± 412	1130 ± 582	3362 ± 104
CDS-PPO	*Hopper*	0.10	3181 ± 47	3125 ± 173	2458 ± 233	1142 ± 59	1846 ± 376
		0.20	3278 ± 36	$\mathbf{3318 \pm 406}$	$\mathbf{3044 \pm 350}$	1723 ± 168	$\mathbf{3044 \pm 131}$
		0.30	$\mathbf{3409 \pm 82}$	2850 ± 112	2797 ± 180	$\mathbf{1983 \pm 228}$	2583 ± 425

The results show that CDS remains effective when the test-time attack budget differs from the training smoothing level, but an overly small σ provides insufficient protection against stronger attacks. A moderate value generally gives the best robustness–performance trade-off: $\sigma = 0.10$ performs best overall on Pong and Freeway, while $\sigma = 0.05$ is preferable for RoadRunner in our main setting. For CDS-PPO, $\sigma = 0.20$ provides the strongest aggregate performance across Walker and Hopper, although the best value can vary by attack and environment. These results support using an environment-dependent smoothing level rather than a single universally optimal value.

D.3 DETAILED EXPERIMENTAL RESULTS OF CERTIFIED RADIUS

Table 5 shows the certified radii of our CDS agents, S-DQNs, and CROP agents. CDS consistently achieves the strongest certified robustness across all three environments. Even without adversarial training, CDS-VanillaDQN markedly enlarges the certified radius over Vanilla S-DQN (and avoids CROP's near-zero radius), demonstrating that reverse-diffusion denoising enables strong plug-and-play certified robustness for non-robust base policies. We use the same sample number of $m = 100$ as S-DQNs Sun et al. (2024).

Table 5: Certified radius of different smoothed DQN agents.

Methods	Certified radius (larger is better)		
	Pong	Freeway	RoadRunner
Ours (CDS, action-smoothed policy):			
CDS-RadialDQN	<u>0.1153</u>	**0.1161**	**0.0581**
CDS-SADQN	**0.1163**	<u>0.1146</u>	<u>0.0566</u>
CDS-VanillaDQN	0.1018	0.0992	0.0541
S-DQNs Sun et al. (2024) (hard randomized smoothing):			
S-DQN (RadialDQN)	0.1024	0.1134	0.0545
S-DQN (S-PGD)	0.0506	0.0735	0.0511
S-DQN (Vanilla)	0.0612	0.0701	0.0497
CROP Wu et al. (2022a) (mean smoothing):			
RadialDQN+RS	0.0000	0.0000	0.0000
SADQN+RS	0.0618	0.0663	0.0000
VanillaDQN+RS	0.0000	0.0000	0.0000

D.4 DETAILED EXPERIMENTAL RESULTS OF ACTION DIVERGENCE

Following Sun et al. Sun et al. (2024), we use Action Divergence (ADIV) to evaluate the certified action stability of smoothed PPO agents. ADIV is computed from the normalized width of the certified action interval,

$$\mathrm{ADIV} = \mathbb{E}_{s,\varepsilon}\left[\frac{\left\|\tilde{\pi}_{\mathrm{det},\overline{p}_{\mathrm{est}}}(s) - \tilde{\pi}_{\mathrm{det},\underline{p}_{\mathrm{est}}}(s)\right\|_2}{2\varepsilon}\right], \tag{77}$$

where ε is the ℓ_2 perturbation budget used to estimate the certified action bound. A lower ADIV indicates a tighter action interval and thus a more stable smoothed policy under ℓ_2-bounded perturbations. As shown in Table 6, CDS-PPO generally achieves lower ADIV than S-PPO and RS+PPO, especially on Hopper, demonstrating that the diffusion-based smoothing procedure yields tighter certified action bounds.

Table 6: Action Divergence (ADIV) of smoothed PPO agents on Walker and Hopper. Lower values indicate tighter certified action bounds.

***Walker*:**			
Base agent	CDS-PPO	S-PPO	RS+PPO
Vanilla	**2.59**	2.91	5.12
Radial	8.11	8.64	**7.97**
WocaR	1.20	**1.15**	2.41
SGLD	**1.25**	1.41	2.23
ATLA	**2.88**	4.20	6.03
PA-ATLA	**3.19**	3.90	5.61
***Hopper*:**			
Base agent	CDS-PPO	S-PPO	RS+PPO
Vanilla	**1.41**	1.60	4.19
Radial	**1.51**	2.33	2.78
WocaR	**0.71**	0.79	1.53
SGLD	**0.50**	0.64	1.36
ATLA	**6.22**	8.96	17.01
PA-ATLA	**3.84**	8.42	11.18

D.5 DETAILED EXPERIMENTAL RESULTS OF REWARD LOWER BOUND

D.5.1 DETAILED EXPERIMENTAL RESULTS OF REWARD LOWER BOUND FOR DQN

Table 7, Table 8 and Table 9 show the reward lower bound of smoothed CDS-DQN agents under ℓ_2 budget ε in the *Pong*, *Freeway*, and *RoadRunner* Atari environments. We set the length of the trajectory $H = 4000$.

Table 7: The reward lower bound of smoothed DQN agents under different ℓ_2 attack budgets. The smoothing variance σ is set to 0.1 in Pong.

Pong	ℓ_2 attack budget				
$\varepsilon(\ell_2)$	0.001	0.002	0.003	0.004	0.005
Ours (CDS):					
CDS-RadialDQN	**21.0**	20.0	20.0	19.0	19.0
CDS-CARDQN	**21.0**	20.0	20.0	19.0	19.0
CDS-SADQN	**21.0**	**21.0**	**21.0**	**21.0**	**21.0**
CDS-WocaRDQN	20.0	19.0	19.0	18.0	18.0
CDS-VanillaDQN	**21.0**	20.0	19.8	19.0	19.0
SDQNs:					
S-DQN (RadialDQN)	20.0	20.0	19.0	18.0	18.0
S-DQN (CARDQN)	20.0	20.0	19.0	19.0	18.4
S-DQN (SADQN)	**21.0**	20.0	20.0	19.0	19.0
S-DQN (WocaR)	19.2	18.0	17.0	12.7	12.0
S-DQN (Vanilla)	18.0	17.0	16.0	15.0	14.0
Randomized Smoothing (RS):					
RadialDQN+RS	-21.0	-21.0	-21.0	-21.0	-21.0
CARDQN+RS	-21.0	-21.0	-21.0	-21.0	-21.0
SADQN+RS	-21.0	-21.0	-21.0	-21.0	-21.0
WocaRDQN+RS	-21.0	-21.0	-21.0	-21.0	-21.0
VanillaDQN+RS	-21.0	-21.0	-21.0	-21.0	-21.0

Table 8: The reward lower bound of smoothed DQN agents under different ℓ_2 attack budgets. The smoothing variance σ is set to 0.1 in Freeway.

Freeway	ℓ_2 attack budget				
$\varepsilon(\ell_2)$	0.001	0.002	0.003	0.004	0.005
Ours (CDS):					
CDS-RadialDQN	**33.0**	**33.0**	**32.0**	**32.4**	**32.0**
CDS-CARDQN	32.0	31.0	31.0	30.7	29.5
CDS-SADQN	31.0	30.0	30.0	29.6	29.0
CDS-WocaRDQN	32.0	31.0	31.0	30.0	30.0
CDS-VanillaDQN	**33.0**	32.0	**32.0**	31.0	31.0
SDQNs:					
S-DQN (RadialDQN)	30.0	30.0	29.0	28.0	28.0
S-DQN (CARDQN)	29.0	28.0	28.0	26.0	25.4
S-DQN (SADQN)	28.0	28.0	27.0	26.7	26.0
S-DQN (WocaR)	28.0	28.0	27.0	25.7	25.0
S-DQN (Vanilla)	31.0	30.0	29.0	29.0	28.0
Randomized Smoothing (RS):					
RadialDQN+RS	20.0	20.0	20.0	20.0	19.0
CARDQN+RS	20.0	20.0	20.0	20.0	19.2
SADQN+RS	20.0	20.0	20.0	20.0	19.0
WocaRDQN+RS	20.0	20.0	20.0	20.0	19.0
VanillaDQN+RS	13.0	12.0	11.0	10.0	9.0

Table 9: The reward lower bound of smoothed DQN agents under different ℓ_2 attack budgets. The smoothing variance σ is set to 0.05 in RoadRunner.

RoadRunner	ℓ_2 attack budget				
$\varepsilon(\ell_2)$	0.001	0.002	0.003	0.004	0.005
Ours (CDS):					
CDS-RadialDQN	**43596**	**37378**	33370	**32754**	**29759**
CDS-CARDQN	38139	34871	**34270**	30525	27921
CDS-SADQN	39127	34556	32120	29588	25681
CDS-WocaRDQN	27896	25435	21590	18185	7115
CDS-VanillaDQN	37248	32849	30490	26344	24390
SDQNs:					
S-DQN (RadialDQN)	36200	29400	21612	14163	14001
S-DQN (CARDQN)	37424	33761	29193	18151	11988
S-DQN (SADQN)	38024	34647	31245	27987	25211
S-DQN (WocaR)	28724	26049	24291	20660	12816
S-DQN (Vanilla)	32215	25097	21123	18067	18001
Randomized Smoothing (RS):					
RadialDQN+RS	9400	5497	2295	2104	2100
CARDQN+RS	900	900	900	890	872
SADQN+RS	17900	15197	12623	9567	9501
WocaRDQN+RS	3300	1200	593	306	300
VanillaDQN+RS	500	100	0	0	0

D.5.2 Detailed experimental results of reward lower bound for PPO

Table 10 shows the certified reward lower bounds for smoothed PPO agents under different ℓ_2 budgets ε. Following the same evaluation protocol, we use a fixed budget ε for every state along the trajectory, which gives a total perturbation budget $B = \varepsilon\sqrt{H}$, where H is the trajectory length. We set $H = 1000$ for both Walker and Hopper. Our CDS-PPOs achieve higher certified reward lower bounds than their smoothed counterparts.

D.6 Detailed experimental results of robust reward

D.6.1 Detailed experimental results of robust reward for DQN

Table 11, Table 12 and Table 13 report the rewards of DQN agents under ℓ_∞ attacks in the *Pong*, *Freeway*, and *RoadRunner* Atari environments. Additional MinBest and PA-AD results on *Pong* and *Freeway* are reported in Table 20. Our CDS agents (action-smoothed) outperform all robust methods across these benchmarks. Specifically, our CDS-VanillaDQN agent shows strong robustness on par with other adversarially trained methods. CDS not only outperforms other methods under low attack strengths ($\varepsilon \leq 0.05$), but also remains robust under higher attack strengths ($0.05 < \varepsilon \leq 0.1$): while the performance of other robust methods degrades rapidly as ε increases, CDS consistently maintains strong and stable performance.

Table 10: The reward lower bound of smoothed PPO agents under different ℓ_2 attack budgets. The smoothing variance σ is set to 0.2 in all environments.

Walker	ℓ_2 attack budget				
$\varepsilon(\ell_2)$	0.002	0.004	0.006	0.008	0.01
Ours (CDS):					
CDS-PPO (Radial)	4283.8	3621.6	2719.0	2170.3	1720.1
CDS-PPO (WocaR)	4177.6	4165.4	4154.4	4144.4	4134.8
CDS-PPO (PA-ATLA)	**6083.1**	**5947.0**	4631.9	3780.4	3159.4
CDS-PPO (SGLD)	4779.5	4764.0	**4748.3**	**4734.8**	**4716.0**
CDS-PPO (Vanilla)	4726.8	4697.3	4667.4	4596.8	3289.6
S-PPOs:					
S-PPO (Radial)	1648	1413	1159	817	550
S-PPO (WocaR)	4345	4333	4322	4308	4296
S-PPO (PA-ATLA)	4364	4017	2526	1573	1100
S-PPO (SGLD)	4496	4478	4460	4440	4420
S-PPO (Vanilla)	4585	4531	4476	4368	2189
Randomized Smoothing (RS+PPOs):					
RadialPPO+RS	1160	987	821	654	420
WocaRPPO+RS	4235	4195	4130	3969	2178
PA-ATLAPPO+RS	606	512	455	416	385
SGLDPPO+RS	4159	3703	2886	2236	1839
VanillaPPO+RS	1263	979	853	748	657
Hopper	ℓ_2 attack budget				
$\varepsilon(\ell_2)$	0.002	0.004	0.006	0.008	0.01
Ours (CDS):					
CDS-PPO (Radial)	2266.9	2060.5	1930.3	1810.5	1714.9
CDS-PPO (WocaR)	**3070.6**	**2451.7**	**2256.2**	**2116.5**	**1981.3**
CDS-PPO (PA-ATLA)	2905.9	2315.2	1784.4	1441.7	1097.3
CDS-PPO (SGLD)	3035.7	2964.8	2386.7	1913.1	1705.9
CDS-PPO (Vanilla)	2539.4	2105.3	1981.1	1860.9	1510.5
S-PPOs:					
S-PPO (Radial)	2865	2294	1925	1760	1574
S-PPO (WocaR)	1691	1573	1470	1397	1360
S-PPO (PA-ATLA)	1883	1603	1438	1309	1176
S-PPO (SGLD)	2783	2758	2732	2710	2661
S-PPO (Vanilla)	1959	1646	1447	1321	1206
Randomized Smoothing (RS+PPOs):					
RadialPPO+RS	2073	1724	1479	1278	1146
WocaRPPO+RS	2076	1832	1696	1533	1473
PA-ATLAPPO+RS	1750	1500	1319	1114	1040
SGLDPPO+RS	1773	1534	1464	1361	1212
VanillaPPO+RS	1300	1218	1046	970	895

Table 11: Rewards of DQN agents under ℓ_∞ PGD attacks in Pong. The smoothing variance σ for the smoothed agents is set to 0.1. Values are rounded to one decimal place before display.

Pong	PGD attack				
ε (ℓ_∞)	0.01	0.02	0.03	0.04	0.05
Ours (CDS):					
CDS-RadialDQN	**21.0 ± 0.7**	**21.0 ± 0.7**	**21.0 ± 0.7**	**21.0 ± 0.7**	**21.0 ± 0.7**
CDS-CARDQN	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**
CDS-SADQN	**21.0 ± 0.4**	**21.0 ± 0.4**	**21.0 ± 0.4**	**21.0 ± 0.4**	**21.0 ± 0.4**
CDS-WocaRDQN	20.0 ± 0.9	20.2 ± 0.7	20.2 ± 0.7	20.0 ± 1.0	19.8 ± 0.7
CDS-VanillaDQN	**21.0 ± 0.1**	20.8 ± 0.2	20.2 ± 0.1	**21.0 ± 0.1**	20.4 ± 0.1
SDQNs:					
S-DQN (RadialDQN)	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	20.6 ± 0.0
S-DQN (CARDQN)	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**
S-DQN (SADQN)	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	20.8 ± 0.4
S-DQN (WocaR)	20.0 ± 1.3	19.4 ± 2.0	19.8 ± 1.6	19.2 ± 2.2	20.0 ± 1.5
S-DQN (Vanilla)	**21.0 ± 0.0**	20.5 ± 0.4	20.2 ± 0.8	17.6 ± 1.8	14.2 ± 4.6
Strong robust baselines:					
RadialDQN	**21.0 ± 0.0**	20.0 ± 2.8	-19.6 ± 0.0	-20.1 ± 0.0	-21.0 ± 0.0
CARDQN	**21.0 ± 0.0**	**21.0 ± 0.0**	-11.2 ± 16.0	-21.0 ± 0.0	-21.0 ± 0.0
SADQN	**21.0 ± 0.0**	-18.6 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
WocaRDQN	19.6 ± 1.0	-19.6 ± 0.0	-20.4 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
VanillaDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
Randomized Smoothing (RS):					
RadialDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
CARDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
SADQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
WocaRDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
VanillaDQN+RS	-20.8 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
ε (ℓ_∞)	0.06	0.07	0.08	0.09	0.1
Ours (CDS):					
CDS-RadialDQN	20.6 ± 0.3	20.3 ± 1.2	20.8 ± 0.7	20.5 ± 0.7	20.0 ± 0.7
CDS-CARDQN	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**
CDS-SADQN	20.6 ± 0.4	**21.0 ± 0.4**	**21.0 ± 0.4**	**21.0 ± 0.4**	**21.0 ± 0.4**
CDS-WocaRDQN	20.2 ± 0.7	20.5 ± 0.2	20.3 ± 0.8	20.0 ± 1.0	19.4 ± 1.5
CDS-VanillaDQN	17.3 ± 0.1	16.6 ± 0.1	14.4 ± 0.1	17.4 ± 0.1	16.0 ± 0.1
SDQNs:					
S-DQN (RadialDQN)	19.2 ± 1.6	18.0 ± 2.7	17.5 ± 3.1	16.0 ± 4.0	16.8 ± 3.6
S-DQN (CARDQN)	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**
S-DQN (SADQN)	20.6 ± 0.5	20.4 ± 0.8	20.0 ± 0.6	20.4 ± 0.5	19.8 ± 0.8
S-DQN (WocaR)	19.4 ± 2.3	19.6 ± 1.7	19.6 ± 1.5	20.0 ± 1.5	19.4 ± 1.9
S-DQN (Vanilla)	-9.3 ± 0.0	-17.9 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
Strong robust baselines:					
RadialDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
CARDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
SADQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
WocaRDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
VanillaDQN	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
Randomized Smoothing (RS):					
RadialDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
CARDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
SADQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
WocaRDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0
VanillaDQN+RS	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0	-21.0 ± 0.0

Table 12: Rewards of DQN agents under ℓ_∞ PGD attacks in Freeway. The smoothing variance σ for the smoothed agents is set to 0.1.

Freeway	PGD attack				
ε (ℓ_∞)	0.01	0.02	0.03	0.04	0.05
Ours (CDS):					
CDS-RadialDQN	33.0 ± 0.0	**33.2 ± 0.4**	**32.6 ± 0.1**	**33.0 ± 0.0**	**33.0 ± 0.0**
CDS-CARDQN	**33.3 ± 0.4**	33.0 ± 0.0	32.5 ± 0.0	32.0 ± 0.0	31.4 ± 0.0
CDS-SADQN	33.0 ± 0.0	**33.2 ± 0.4**	**32.6 ± 0.1**	**33.0 ± 0.0**	**33.0 ± 0.0**
CDS-WocaRDQN	32.4 ± 0.5	32.4 ± 0.5	**32.6 ± 0.5**	32.8 ± 0.5	32.8 ± 0.2
CDS-VanillaDQN	32.0 ± 0.0	32.2 ± 0.0	32.4 ± 0.2	32.2 ± 0.1	32.0 ± 0.2
SDQNs:					
S-DQN (RadialDQN)	32.8 ± 0.4	32.6 ± 1.0	**32.6 ± 1.0**	32.4 ± 0.8	32.2 ± 1.0
S-DQN (CARDQN)	33.0 ± 0.6	33.0 ± 0.6	32.4 ± 1.0	31.6 ± 1.3	31.4 ± 0.5
S-DQN (SADQN)	30.0 ± 0.0	30.0 ± 0.0	30.2 ± 0.4	30.2 ± 0.4	29.4 ± 0.8
S-DQN (WocaR)	30.8 ± 1.3	30.8 ± 1.2	31.0 ± 1.1	30.8 ± 1.3	30.4 ± 0.5
S-DQN (Vanilla)	32.0 ± 0.4	30.0 ± 1.0	20.4 ± 1.0	0.0 ± 0.0	0.0 ± 0.0
Strong robust baselines:					
RadialDQN	33.0 ± 0.0	28.4 ± 1.2	22.8 ± 1.9	20.0 ± 1.1	21.0 ± 0.6
CARDQN	32.0 ± 1.6	27.8 ± 0.7	20.4 ± 0.7	21.0 ± 1.0	18.2 ± 1.6
SADQN	30.0 ± 0.0	27.2 ± 1.2	20.4 ± 0.5	20.8 ± 1.0	18.8 ± 1.3
WocaRDQN	29.0 ± 1.3	21.8 ± 2.0	20.6 ± 0.8	21.2 ± 1.0	22.0 ± 0.0
VanillaDQN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Randomized Smoothing (RS):					
RadialDQN+RS	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0
CARDQN+RS	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2	22.6 ± 1.6	23.2 ± 2.7
SADQN+RS	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	21.6 ± 0.0
WocaRDQN+RS	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0	22.2 ± 0.0
VanillaDQN+RS	22.2 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
ε (ℓ_∞)	0.06	0.07	0.08	0.09	0.1
Ours (CDS):					
CDS-RadialDQN	**32.8 ± 0.2**	**32.8 ± 1.3**	32.6 ± 0.8	32.4 ± 0.8	**33.0 ± 0.7**
CDS-CARDQN	31.6 ± 0.0	30.8 ± 0.0	30.8 ± 1.1	30.6 ± 0.8	30.2 ± 1.5
CDS-SADQN	**32.8 ± 0.2**	32.6 ± 1.3	32.6 ± 0.8	32.4 ± 0.8	**33.0 ± 0.7**
CDS-WocaRDQN	32.2 ± 0.6	32.2 ± 0.5	**32.8 ± 0.6**	32.5 ± 0.5	32.2 ± 0.7
CDS-VanillaDQN	32.0 ± 0.4	32.6 ± 0.5	32.2 ± 0.4	**32.6 ± 0.8**	32.5 ± 0.8
SDQNs:					
S-DQN (RadialDQN)	**32.8 ± 0.7**	31.4 ± 1.3	30.8 ± 0.7	30.6 ± 0.8	30.8 ± 1.3
S-DQN (CARDQN)	30.4 ± 1.4	30.8 ± 1.7	30.2 ± 1.0	28.0 ± 0.9	27.6 ± 1.5
S-DQN (SADQN)	29.4 ± 0.8	30.8 ± 1.0	28.2 ± 0.7	27.0 ± 1.4	26.2 ± 1.9
S-DQN (WocaR)	29.2 ± 0.8	30.0 ± 1.3	26.2 ± 1.9	26.8 ± 0.7	26.6 ± 1.4
S-DQN (Vanilla)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Strong robust baselines:					
RadialDQN	20.4 ± 1.4	20.8 ± 0.3	20.6 ± 1.2	20.6 ± 0.4	20.6 ± 0.4
CARDQN	16.4 ± 0.8	13.4 ± 2.3	12.4 ± 0.4	10.6 ± 0.8	7.2 ± 0.3
SADQN	19.4 ± 0.7	20.8 ± 1.3	21.4 ± 0.4	20.0 ± 1.6	19.2 ± 1.7
WocaRDQN	22.6 ± 1.0	22.0 ± 0.8	21.4 ± 1.9	18.8 ± 0.9	19.6 ± 1.2
VanillaDQN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Randomized Smoothing (RS):					
RadialDQN+RS	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2
CARDQN+RS	21.8 ± 0.7	22.0 ± 1.0	21.6 ± 0.8	21.2 ± 0.9	19.6 ± 2.0
SADQN+RS	21.4 ± 1.3	21.8 ± 1.4	22.2 ± 1.1	22.2 ± 1.7	22.4 ± 1.3
WocaRDQN+RS	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2	22.2 ± 2.2
VanillaDQN+RS	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 13: Rewards of DQN agents under ℓ_∞ PGD attacks in RoadRunner. The smoothing variance σ for the smoothed agents is set to 0.05.

RoadRunner	PGD attack				
ε (ℓ_∞)	0.01	0.02	0.03	0.04	0.05
Ours (CDS):					
CDS-RadialDQN	43680 ± 7176	39440 ± 5904	**42680 ± 6696**	**34240 ± 7296**	**29140 ± 4993**
CDS-CARDQN	41220 ± 4847	34640 ± 14081	23000 ± 16281	18520 ± 13895	19180 ± 9044
CDS-SADQN	**45960 ± 7406**	**44700 ± 9184**	31700 ± 4374	21480 ± 3134	17100 ± 1577
CDS-WocaRDQN	32440 ± 7026	24580 ± 5002	25100 ± 2202	20720 ± 2235	17420 ± 2388
CDS-VanillaDQN	29320 ± 3841	21360 ± 5963	15140 ± 3373	9500 ± 2913	4100 ± 3381
SDQNs:					
S-DQN (RadialDQN)	38643 ± 4340	33100 ± 4532	39230 ± 2870	31060 ± 6920	27220 ± 7912
S-DQN (CARDQN)	37860 ± 13763	34500 ± 17078	16460 ± 8088	13800 ± 9279	9274 ± 8264
S-DQN (SADQN)	41920 ± 5753	38600 ± 5736	34320 ± 7240	22660 ± 2193	17480 ± 1345
S-DQN (WocaR)	30300 ± 7874	23100 ± 3830	25460 ± 7889	18840 ± 1756	6333 ± 8964
S-DQN (Vanilla)	28750 ± 3784	16480 ± 3416	460 ± 382	0 ± 0	0 ± 0
Strong robust baselines:					
RadialDQN	43520 ± 4081	24160 ± 2604	15500 ± 6466	1020 ± 393	620 ± 492
CARDQN	42980 ± 10970	13620 ± 8466	1940 ± 1155	3420 ± 2071	640 ± 372
SADQN	28580 ± 2584	3240 ± 1544	780 ± 840	420 ± 523	100 ± 200
WocaRDQN	1580 ± 2108	0 ± 0	0 ± 0	0 ± 0	0 ± 0
VanillaDQN	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Randomized Smoothing (RS):					
RadialDQN+RS	11260 ± 0	9220 ± 0	6680 ± 0	7780 ± 0	3180 ± 0
CARDQN+RS	960 ± 48	960 ± 48	960 ± 48	960 ± 48	960 ± 48
SADQN+RS	14240 ± 0	16440 ± 0	1960 ± 0	1040 ± 0	930 ± 0
WocaRDQN+RS	560 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
VanillaDQN+RS	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
ε (ℓ_∞)	0.06	0.07	0.08	0.09	0.1
Ours (CDS):					
CDS-RadialDQN	**25840 ± 2595**	**19300 ± 2376**	**17840 ± 833**	9440 ± 6747	9120 ± 6506
CDS-CARDQN	14640 ± 7678	7860 ± 4210	6780 ± 4130	5060 ± 2605	3420 ± 479
CDS-SADQN	14740 ± 2048	11620 ± 2718	7880 ± 3410	8640 ± 2384	4640 ± 4081
CDS-WocaRDQN	17520 ± 4075	12320 ± 5723	4875 ± 2537	3460 ± 2273	2897 ± 1863
CDS-VanillaDQN	4800 ± 3465	0 ± 0	0 ± 0	0 ± 0	0 ± 0
SDQNs:					
S-DQN (RadialDQN)	20300 ± 3686	17320 ± 1387	12400 ± 4825	**10880 ± 5945**	**10800 ± 2929**
S-DQN (CARDQN)	6980 ± 4220	5381 ± 2840	4180 ± 1147	3741 ± 1723	3000 ± 1093
S-DQN (SADQN)	8640 ± 3389	7540 ± 2137	7580 ± 2330	5820 ± 4006	4900 ± 729
S-DQN (WocaR)	550 ± 777	0 ± 0	0 ± 0	0 ± 0	0 ± 0
S-DQN (Vanilla)	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Strong robust baselines:					
RadialDQN	580 ± 462	840 ± 968	80 ± 74	280 ± 391	220 ± 285
CARDQN	1340 ± 981	660 ± 685	1220 ± 915	2380 ± 617	1140 ± 1174
SADQN	20 ± 40	0 ± 0	0 ± 0	0 ± 0	0 ± 0
WocaRDQN	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
VanillaDQN	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Randomized Smoothing (RS):					
RadialDQN+RS	4620 ± 2081	3660 ± 1111	3500 ± 2024	1860 ± 941	1760 ± 338
CARDQN+RS	960 ± 48	960 ± 48	960 ± 48	960 ± 48	960 ± 48
SADQN+RS	800 ± 551	600 ± 807	600 ± 536	600 ± 579	660 ± 880
WocaRDQN+RS	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
VanillaDQN+RS	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0

D.6.2 DETAILED EXPERIMENTAL RESULTS OF ROBUST REWARD FOR PPO

Table 14: Rewards of PPO agents under different adversarial attacks in Walker and Hopper. The smoothing variance σ for the smoothed agents is set to 0.2. The ℓ_∞ attack budget is set to 0.075 in both environments.

Walker	Clean reward	MAD attack	Min-RS attack	Optimal attack	PA-AD attack
Ours (CDS):					
CDS-PPO (SGLD)	4822	4785	4728	2206	4754
CDS-PPO (Radial)	4029	3978	3698	3539	2951
CDS-PPO (ATLA)	4667	4509	2894	2018	2080
CDS-PPO (PA-ATLA)	**5808**	**5775**	**5227**	3049	**5321**
CDS-PPO (WocaR)	4918	4175	4729	**4263**	4117
CDS-PPO (Vanilla)	4752	4709	4596	2368	4306
SPPOs:					
S-PPO (SGLD)	4566	4537	4241	4085	4026
S-PPO (Radial)	2117	2160	1028	915	689
S-PPO (WocaR)	4363	4360	3907	3920	3867
S-PPO (S-ATLA)	4897	4460	2170	4230	2980
S-PPO (S-PA-ATLA)	4407	4045	2379	144	372
S-PPO (Vanilla)	4552	4386	3203	944	1077
Strong robust baselines:					
SGLDPPO	4329	4177	2376	2747	718
RadialPPO	2221	2230	1270	132	152
WocaRPPO	4110	3918	1950	2916	2067
ATLAPPO	3564	2567	672	818	263
PA-ATLAPPO	2548	1717	591	183	298
VanillaPPO	4301	2806	551	437	275
Randomized Smoothing (RS):					
SGLDPPO+RS	4290	4124	2739	1615	717
RadialPPO+RS	1804	1883	610	145	208
WocaRPPO+RS	4013	4160	1362	3211	1765
ATLAPPO+RS	4129	3348	894	1090	322
PA-ATLAPPO+RS	1325	1990	427	322	332
VanillaPPO+RS	3582	2892	592	440	401
Hopper					
Ours (CDS):					
CDS-PPO (SGLD)	3085	3056	3011	2274	3067
CDS-PPO (Radial)	2280	3086	2676	1962	2211
CDS-PPO (ATLA)	3752	**3713**	**3513**	**2820**	3738
CDS-PPO (PA-ATLA)	3538	3230	3350	2035	3205
CDS-PPO (WocaR)	3520	3407	3426	2617	**3873**
CDS-PPO (Vanilla)	3278	3318	3044	1723	3044
SPPOs:					
S-PPO (SGLD)	2894	2896	2428	1579	1523
S-PPO (Radial)	3756	3205	1212	1285	2015
S-PPO (WocaR)	2335	2194	1328	1053	1189
S-PPO (S-ATLA)	**3770**	2557	1752	2595	1927
S-PPO (S-PA-ATLA)	3737	2631	1839	1655	1950
S-PPO (Vanilla)	3583	2765	1049	995	1190
Strong robust baselines:					
SGLDPPO	2772	2587	1107	1087	1463
RadialPPO	3291	3056	1182	900	1161
WocaRPPO	3652	2993	1111	1112	1331
ATLAPPO	3577	1493	1245	1172	1124
PA-ATLAPPO	3508	3297	1110	1518	1842
VanillaPPO	3321	2375	834	695	789
Randomized Smoothing (RS):					
SGLDPPO+RS	2354	2386	1106	1059	1411
RadialPPO+RS	3298	2876	1011	1122	1165
WocaRPPO+RS	3535	2878	1084	1095	1208
ATLAPPO+RS	3278	1485	1220	1161	1129
PA-ATLAPPO+RS	3537	3027	1365	1861	1866
VanillaPPO+RS	3211	2238	920	707	840

D.7 ADDITIONAL EXPERIMENTAL RESULTS

Table 15: Certified ratio of smoothed DQN agents under action-level certification. Our CDS-DQNs consistently achieve a higher certified ratio η_r than S-DQNs.

Model/metric	Certified Ratio of *Pong*			Certified Ratio of *Freeway*		
Certified radius r	0.03	0.06	1.0	0.03	0.06	1.0
S-DQNs Sun et al. (2024)						
S-DQN (CAR) Li et al. (2024)	97.6	95.1	80.8	96.1	94.1	81.5
S-DQN (Radial) Oikarinen et al. (2021)	95.8	91.6	62.4	99.4	98.4	90.8
S-DQN (SADQN) Zhang et al. (2020)	**100.0**	**100.0**	93.8	99.7	98.2	94.7
S-DQN (WocaR) Liang et al. (2022)	96.8	94.3	80.2	97.2	95.6	91.4
S-DQN (Vanilla)	65.5	56.8	36.2	85.8	72.7	46.7
Our Methods:						
CDS-DQN (CAR) Li et al. (2024)	**100.0**	**100.0**	**100.0**	99.3	98.9	97.7
CDS-DQN (Radial) Oikarinen et al. (2021)	96.7	91.9	74.5	99.9	99.8	99.7
CDS-DQN (SADQN) Zhang et al. (2020)	**100.0**	**100.0**	96.2	**100.0**	**100.0**	**100.0**
CDS-DQN (WocaR) Liang et al. (2022)	96.2	95.6	83.7	99.5	98.5	94.7
CDS-DQN (Vanilla)	99.7	99.2	96.5	93.0	86.5	67.9

Table 16: Certified radius of smoothed DQN agents under value-based certification.

	Certified radius	Smoothing variance		
	σ	0.005	0.05	0.1
***Pong*:**	CDS-DQN (Vanilla)	0.0033	0.0245	0.0315
	CDS-DQN (Radial)	0.0044	0.0395	0.0790
	CDS-DQN (SADQN)	**0.0062**	**0.0583**	**0.1151**
	VanillaDQN	0.0009	0.0035	0.0000
	RadialDQN	0.0003	0.0350	0.0735
	SADQN	0.0019	0.0352	0.0817
	VanillaDQN+RS	0.0004	0.0000	0.0000
***Freeway*:**	CDS-DQN (Vanilla)	0.0041	0.0386	0.0763
	CDS-DQN (Radial)	0.0054	0.0418	0.1028
	CDS-DQN (SADQN)	**0.0058**	**0.0582**	**0.1163**
	VanillaDQN	0.0013	0.0062	0.0086
	RadialDQN	0.0021	0.0117	0.0473
	SADQN	0.0058	0.0553	0.1005
	VanillaDQN+RS	0.0005	0.0000	0.0000

Table 19: Comparison of test-time runtime (FPS) in the clean environment.

Pong	SA-DQN	WocaR-DQN	Radial-DQN	S-DQN	CDS-DQN (Ours)
Testing time (FPS)	502	635	422	64	14.5
Hopper	RadialPPO	PA-ATLAPPO	WocaRPPO	S-PPO	CDS-PPO (Ours)
Testing time (FPS)	826	1514	1335	258	139

Table 17: Robust reward of CDS-VanillaDQN (action-smoothed) under ℓ_∞ PGD attacks with different smoothing variance σ.

Pong	PGD attack				
$\varepsilon(\ell_\infty)$	0.01	0.02	0.03	0.04	0.05
CDS-VanillaDQN ($\sigma = 0.01$)	19.8 ± 1.5	11.4 ± 7.1	-19.0 ± 2.0	-19.6 ± 1.3	-20.8 ± 0.3
CDS-VanillaDQN ($\sigma = 0.05$)	20.0±1.5	20.6±0.8	19.8±1.4	19.0±2.0	19.0±2.0
CDS-VanillaDQN ($\sigma = 0.1$)	**21.0 ± 0.1**	**20.8 ± 0.2**	**20.2 ± 0.1**	**21.0 ± 0.1**	**20.4 ± 0.1**
CDS-VanillaDQN ($\sigma = 0.15$)	16.8±2.1	15.0±2.8	16.0±3.5	14.8±2.2	15.6±2.7
Freeway					
CDS-VanillaDQN ($\sigma = 0.01$)	11.4±1.4	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
CDS-VanillaDQN ($\sigma = 0.05$)	32.2±0.7	**32.8±1.1**	**32.8±0.9**	**32.6±0.4**	**32.6±0.5**
CDS-VanillaDQN ($\sigma = 0.1$)	32.0±0.0	32.2±0.0	32.4±0.2	32.2±0.1	32.0±0.2
CDS-VanillaDQN ($\sigma = 0.15$)	**32.6±1.0**	32.6±1.0	**32.8±0.7**	32.4±1.0	32.6±1.0
RoadRunner					
CDS-VanillaDQN ($\sigma = 0.01$)	4920±4419	0±0	0±0	0±0	0±0
CDS-VanillaDQN ($\sigma = 0.05$)	**29320 ± 3841**	21360 ± 5963	15140 ± 3373	9500 ± 2913	4100 ± 3381
CDS-VanillaDQN ($\sigma = 0.1$)	25640±4350	**26300±3573**	17520±2324	**19600±2488**	15480±1674
CDS-VanillaDQN ($\sigma = 0.15$)	20540±440	**21880±1519**	**21080±2825**	19460±889	**23120±3038**

Table 18: Robust reward of CDS-DQN (Vanilla) under ℓ_2 PGD attacks using action-level and value-level certification. The two certified strategies achieve very similar high robust rewards over the entire ε range.

Pong	PGD attack				
$\varepsilon(\ell_2)$	0.01	0.03	0.05	0.07	0.1
Action-level	**21.0±0.0**	20.0±0.6	**20.4±0.8**	19.6±0.8	**18.2±1.3**
Value-level	**21.0±0.0**	**20.4±0.8**	20.0±1.2	**19.8±0.7**	18.0±1.1
Freeway					
Action-level	**32.0±0.0**	**31.8±0.9**	**31.6±1.0**	30.6±1.0	**28.9±1.4**
Value-level	**32.0±0.0**	31.4±1.1	31.0±0.8	**30.8±0.9**	28.6±1.2
RoadRunner					
Action-level	43680±6679	45920±6368	40040±10272	38700±5306	29800±4447
Value-level	**46100±10386**	**47240±8462**	**43580±10821**	**46300±3901**	**33880±5122**

Table 20: Additional robust reward results of DQN agents under MinBest and PA-AD attacks in *Pong* and *Freeway*. The attack budgets are reported in pixel scale.

Environment	MinBest		PA-AD	
Method	$\varepsilon = 1/255$	$\varepsilon = 3/255$	$\varepsilon = 1/255$	$\varepsilon = 3/255$
***Pong*:**				
SA-DQN	21.0 ± 0.0	21.0 ± 0.0	21.0 ± 0.0	19.1 ± 1.8
RADIAL-DQN	21.0 ± 0.0	21.0 ± 0.0	21.0 ± 0.0	20.0 ± 2.4
WocaR-DQN	21.0 ± 0.0	20.6 ± 0.2	21.0 ± 0.0	19.4 ± 2.2
CDS-DQN (Vanilla) (Ours)	**21.0 ± 0.0**	**21.0 ± 0.0**	**21.0 ± 0.0**	**20.2 ± 2.0**
***Freeway*:**				
SA-DQN	30.0 ± 0.0	30.0 ± 0.0	21.3 ± 4.7	7.3 ± 4.3
RADIAL-DQN	**33.3 ± 0.0**	**33.3 ± 0.0**	19.4 ± 4.0	12.6 ± 4.1
WocaR-DQN	31.2 ± 0.3	31.0 ± 0.3	25.0 ± 4.1	13.8 ± 3.6
CDS-DQN (Vanilla) (Ours)	33.0 ± 0.0	33.0 ± 0.0	**30.0 ± 3.9**	**30.3 ± 2.8**