Appendices

The appendices can be summarised as follows:

- App.A: We provide more details about the Environments used in this paper and smoothed policy training algorithms mentioned in Sections 3.2 and 5.
- App. B: We provide the poof for Corollary 1 mentioned in Section 3.1.
- App. C: We provide proof for Theorem 2 mentioned in Section 4.4.
- App. D: We provide detailed proof to establish the relationship between HS divergence and l_2-norm distance between two distributions (mentioned in Section 4.4).
- App. E, F, G: We present the proof for Theorem 3, 4, 5 mentioned in Sections 4.4, 4.5, and 4.6;
- App. H: In this section, we provide more related work about the adversarial robustness and certification for RL.
- App.I: We introduce additional experiments conducted in the 'Mountain Car' and 'Freeway' environments, along with supplementary analyses.

A Environments and Policy

In this section, we will show how to train the smoothed policy given noise sampling from Gaussian distribution.

A.1 Environment

First, we introduce all environments tested in this paper. All environments are implemented from 'OpenAI gym' (Brockman et al. 2016).

CartPole In the CartPole environment, the state of the environment is defined by a 4-dimensional vector: cart position, cart velocity, pole angle, and pole angular velocity. Two actions can be taken: moving the cart to the left (action 0) and moving the cart to the right (action 1). The agent receives a reward of +1 for each time step in which the pole remains upright. The goal is to keep the pole balanced for as long as possible.

Freeway The game is played on a 2D grid where the player's character starts at the bottom of the screen and needs to cross a multi-lane freeway to reach the top of the screen, which engages high-dimensional observation. The player can take three discrete actions: remain, down, and up. In this paper, our experiments are employed on the Grayscale image observation with 84 * 84 pixels. It plays a 'Hard' mode and terminates after 250 time steps.

Mountain Car 'Mountain Car' environment is played in a continuous space. If the car reaches the destination within in 999steps, it will be rewarded 1 else it gains 0 rewards. The state of the environment is represented by a 2-dimensional vector: car position and velocity, and action is the continuous scalar that represents the acceleration.

A.2 Learning algorithm

There are two reinforcement learning algorithms used in this paper: DDPG (Deep Deterministic Policy Gradient) and DQN (Deep Q-Network). The DDPG is primarily used for solving continuous action space problems, where the action space is not discrete and can take on a wide range of values. This allows the DDPG to be applied in the 'Mountain Car' environment.

DDPG uses a deterministic policy. This means that for a given state, the DDPG algorithm directly outputs the best action to take, rather than producing a probability distribution over actions. On the other hand, DQN works with a Q function that estimates the expected cumulative reward for each action in a given state. It then selects the action with the highest Q-value as the optimal action. As for the training algorithm, DDPG uses a form of actor-critic architecture, where it maintains both an actor-network (which estimates the policy) and a critic network (which estimates the Q-values). The actor-network learns the best actions to take, while the critic network provides feedback on how good these actions are. DQN uses a simpler architecture where it only maintains a Q-network. It learns the Q-values directly from the data generated during training. Below we present the specific training algorithm for a smoothed policy.

B Proof of Corollary 1

Corollary 1. *(Adaptive Generalized Donsker-Varadhan Variational Formula for RL) Suppose the constraint set is defined by f, let $z(\tau) = \frac{q(\tau)}{p(\tau)}$, where $q = \mu(s'_t), p = \mu(s_t)$, and $J(\tau)$ denoting as the cumulative reward of the smoothed policy. Given the convex function f with $f(1) = 0$, we have*

$$\min_{\Gamma} \left\{ D(q||p) + \mathop{\mathrm{E}}_{\tau\sim p}[z(\tau)J(\tau)] \right\} = \max_{\eta\in \mathrm{R}} \left\{ \eta - \mathop{\mathrm{E}}_{\tau\sim p}[f^*(\eta - J(\tau))] \right\}, \tag{1}$$

Algorithm 1: Smoothed Policy training process for DQN

Input: Gaussian distribution parameter σ; weights update step h

function TRAIN SMOOTHED POLICY(M, Q, α, σ)
  Initialise the replay memory D;exploration rate is p
  Initialise the weights of action-value function Q and target action-value function Q_{target}
  for $episode \leftarrow 1, N$ **do**
    Initialise the observation at state s_0 as $s'_0 = s_0 + \Delta_0$ where $\Delta_0 \sim \mathcal{N}(0, \sigma^2 I_N)$.
    for $t \leftarrow 1, T$ **do**
      Choose action a_t from random with probability p or $\arg\max_{a_t \in \mathcal{A}} Q^{\pi}(s_t + \Delta_t, a_t)$ where $\Delta_t \sim \mathcal{N}(0, \sigma^2 I_D)$
      Execute a_t and obtain the reward r and next state s_{t+1}
      Store transition $\tau = (s_t, a_t, r, s_{t+1})$ in D
      $s_t \leftarrow s_{t+1}$
      if Enough experience in D **then**
        Sample random minibatch of transitions from D
        for Each transition i in the minibatch **do**
          $y_i = r_i + \gamma \max_{a_{i+1} \in \mathcal{A}} Q(s_{i+1}, a_{i+1})$
        Calculate the loss $\mathcal{L} = 1/\mathrm{N} \sum_{i=0}^{N-1} \left(Q\left(s_i, a_i\right) - y_i\right)^2$
        Minimising the loss $\mathcal{L}$ by SGD to update Q
        For every h steps, copy weights from Q to the target Q_{target}

Algorithm 2: Smoothed Policy training process for DDPG

Input: Gaussian distribution parameter σ;

function TRAIN SMOOTHED POLICY(M, Q, α, σ)
  Initialise the weights of action-value function Q and target action-value function Q_{target}
  for $episode \leftarrow 1, N$ **do**
    Initialise weights parameter θ.
    Initialise the observation at state s_0 as $s'_0 = s_0 + \Delta_0$ where $\Delta_0 \sim \mathcal{N}(0, \sigma^2 I_N)$.
    while $s \neq$ finite state **do**
      Choose action a_t from random with probability p or $\pi^{\theta}\left(s_t + \Delta_t\right)$ where $\Delta_t \sim \mathcal{N}(0, \sigma^2 I_D)$
      Execute a_t and obtain the reward r and next state s_{t+1}
      Store transition $\tau = (s_t, a_t, r, s_{t+1})$ in D
      if Enough experience in D **then**
        Sample random minibatch of transitions from D
        for Each transition j in the minibatch **do**
          $y_i = r_i + \gamma Q(s_{i+1}, \pi^{\theta}(s_{i+1}))$
        Calculate the loss $\mathcal{L} = 1/\mathrm{N} \sum_{i=0}^{N-1} \left(Q\left(s_i, a_i\right) - y_i\right)^2$
        Minimising the loss $\mathcal{L}$ by SGD to update Q
        Update the weights θ

Proof. First, given the well-defined densities $p(\tau), q(\tau)$, and $p(\tau) > 0$ whenever $q(\tau) > 0$. with Radon Nykodim derivative $z = \frac{q}{p}$, the optimisation goal can be rewritten as $\min_{\tau\sim q}\mathrm{E}\left[J(\tau)\right] = \min_{\tau\sim p}\mathrm{E}\left[z(\tau)J(\tau)\right]$. Constraints on the f-divergence are: $\mathrm{E}_{\tau\sim p}[z(\tau)] = 1$, and $\mathrm{E}_{\tau\sim p}[f(z(\tau)] \leq \epsilon$. Therefore, we can rewrite the optimisation goal as:

$$\min_{\tau\sim p}\left\{\mathrm{E}\left[z(\tau)J(\tau)\right] + \mathrm{E}[f(z(\tau))] \mid \mathrm{E}[z(\tau)] = 1, z(\tau) \geq 0\right\} \tag{2}$$

The Lagrangian dual can be expressed as:

$$\begin{aligned}
&= \max_{\eta\in\mathbb{R}}\left\{\eta + \min_{\tau\sim p}\left\{\mathrm{E}[f(z(\tau))] - \mathrm{E}\left[(\eta - J(\tau))z(\tau)\right]\right\}\right\} \\
&= \max_{\eta\in\mathbb{R}}\left\{\eta - \max_{\tau\sim p}\left\{\mathrm{E}\left[(\eta - J(\tau))z(\tau)\right] - \mathrm{E}\left[f(z(\tau))\right]\right\}\right\} \\
&= \max_{\eta\in\mathbb{R}}\left\{\eta - \mathop{\mathrm{E}}_{\tau\sim p}[\max_{z\geq 0}\{(\eta - J(\tau))z - f(z)\}\right\} \\
&= \max_{\eta\in\mathbb{R}}\left\{\eta - \mathop{\mathrm{E}}_{\tau\sim p}[\eta - J(\tau)]\right\}
\end{aligned}$$

□

C Proof of Theorem 2

Theorem 1. *(Restate the **Theorem 2** in main body) (CDF-based method in optimisation framework) Suppose the reward range is (a, b) and considering n thresholds, denoted as $a < g_1 < g_2 < ... < g_n < b$. Let θ_i represent the lower bound of probability that the total reward obtained by the smoothed policy is above the threshold g_i. Let f* be the convex conjugate of hockey-stick divergences, we have*

$$\begin{aligned}
&\max_{\nu>0,\eta\in\mathbb{R}} \nu\big[\eta - \big(\left(1-\theta_1\right) f^*\left(\eta + \epsilon - \tfrac{a}{\nu}\right) \\
&+ \textstyle\sum_{i=1}^{n}(g_{i+1} - g_i)\theta_i f^*\left(\eta + \epsilon - \tfrac{g_i}{\nu}\right) + \theta_n f^*\left(\eta + \nu\epsilon - \tfrac{g_n}{\nu}\right)\big)\big]
\end{aligned}$$

As in (Kumar, Levine, and Feizi 2021), the lower bound of cumulative reward can be calculated by taking the worst cases with threshold $(g_1, g_2, ..., g_n)$ within the range of (a, b). Referred to (Dvijotham et al. 2020), this problem can be interpreted as an " information-limited" problem, which only uses the information of the probability of whether the output is above a threshold instead of using the full information of the output reward. Let θ_i represent the probability of cumulative reward is above the threshold g_i, $P(J(\tau) \geq g_i)$, we leakage the results in (Kumar et al. 2020) as:

$$\mathop{E}_{\tau\sim p}(J(\tau)) \geq (1-\theta_1)a + \sum_{i=1}^{n}(\theta_i - \theta_{i+1})g_i + \theta_n g_n \tag{3}$$

Noticed that they assume the variable output between $[g_i, g_{i+1})$ to be g_i. Therefore, in our framework, the "information-limited" problem to find the lower bound of mean reward can be expressed as:

$$\max_{\nu>0,\eta\in\mathbb{R}} \nu\big[\eta - \big(\left(1-\theta_1\right) f^*\left(\eta + \epsilon - \frac{a}{\nu}\right) + \sum_{i=1}^{n}(g_{i+1} - g_i)\theta_i f^*\left(\eta + \epsilon - \frac{g_i}{\nu}\right) + \theta_n f^*\left(\eta + \nu\epsilon - \frac{g_n}{\nu}\right)\big)\big]$$

D Robustness certification for information-limited framework

Corollary 2. *(HS Divergency in a function of the l_2 distance) Given the constraint on the l_2 distance, $\mathcal{D}_\epsilon := \{\mu(x') : d(x, x') = |x - x'|_2 \leq \epsilon\}$ and let $\epsilon_\lambda = max_q D_{HS,\lambda}(p||q)$.Then we can define the parameter of the hockey-stick divergence $\mathcal{D}_{HS}$ as the optimal value of following optimisation function:*

$$\lambda^* = \arg\max_{\lambda\geq 0}\left(1 - \lambda\left(1-\theta\right) - \left(\epsilon_\lambda + \max\left(1-\lambda, 0\right)\right)\right) \tag{4}$$

Define the hockey-stick constraint set as $\mathcal{D}_{HS} = \{q : D_{HS,\lambda^*}(q||p) \leq \epsilon_{\lambda^*})\}$.

Suppose the distribution p is a standard norm distribution around x, $p = \mathcal{N}(x, \sigma^2 I_D)$, $q = \mathcal{N}(x', \sigma^2 I_D)$. Given the probability that the output is greater than a threshold $P_{x\sim p}[h(x) = 1] \geq \theta$, where $h(x)$ is the function that indicates whether the reward is greater than the threshold. Let Φ be the CDF of the standard norm distribution. The constraint set $\mathcal{D}_{HS}$ gives us following robustness guarantee:

$$P_{x\sim q}[h(x) = 1] \geq \Phi(\Phi^{-1}(\theta) - \frac{\epsilon}{\sigma}) \tag{5}$$

which is the guarantee of Kumar, Levine, and Feizi (2021).

Proof. Following the proof idea of (Dvijotham et al. 2020), to attain robust certification under any given set of constraints $\mathcal{D}$, it suffices to comprehend the "envelope" of $\mathcal{D}$ concerning all hockey-stick divergences with $\lambda \geq 0$, Therefore, $\lambda^* = \max \mathcal{D}_{HS,\lambda}(q||p)$ captures all the essential information required to offer certification concerning $\mathcal{D}$.

We first consider the optimisation problem,

$$\min_{h(x)\to\{0,1\}} E_{x\sim q}[h(x)] \quad s.t. \quad E_{x\sim p}[1[h(x)=1]] \geq \theta \tag{6}$$

Given $z(x) = \frac{q(x)}{p(x)}$, the dual Laplacian of the optimisation problem is:

$$\begin{aligned}
&\min_{h(x)\to\{0,1\}} \underset{x\sim p}{E}[h(x)z(x)] - \lambda(\underset{x\sim p}{E}[h(x)] - \theta) \\
&= \min_{h(x)\to\{0,1\}} \lambda\theta - \underset{x\sim p}{E}[(z(x)-\lambda)h(x)] \\
&= \lambda\theta + \underset{x\sim p}{E}[\min(z(x)-\lambda, 0)] \\
&= \lambda\theta + \underset{x\sim p}{E}[z(x) - \lambda - \max(z(x)-\lambda, 0)] \\
&= \lambda\theta + E[z(x)-\lambda] - \underset{x\sim p}{E}[\max(z(x)-\lambda, 0)] \\
&= 1 - \lambda(1-\theta) - (D_{HS,\lambda}(q||p) + \max(1-\lambda, 0))
\end{aligned} \tag{7}$$

Therefore, the dual problem can be written as:

$$\arg\max_{\lambda\geq 0} \left(1 - \lambda\left(1-\theta\right) - \left(D_{HS,\lambda}(q||p) + \max\left(1-\lambda, 0\right)\right)\right) \tag{8}$$

under strong duality, the optimal value of the aforementioned problem exactly corresponds to the optimal value of Eq. 6. Therefore, the lower bound of $P_{x\sim q}[h(x)=1]$ can be obtained by solving the dual problem.

For the Gaussian smoothing measure used in (Kumar, Levine, and Feizi 2021), adopting Eq. 4, we can compute the hockey-stick divergence for $\lambda \geq 0$:

$$\max_{x':d(x,x')\leq\epsilon} D_{\mathrm{HS},\lambda}\left(\mu\left(x'\right)\|\mu(x)\right), \tag{9}$$

where $\mu(x) = \mathcal{N}(x, \sigma^2 I_D)$. Here we introduce the result of (Balle and Wang 2018) to help us to complete the proof of Corollary 2.

Theorem 2. *Given the CDF of the standard normal $\mathcal{N}(0,1)$, Φ, $\forall\lambda \geq 0$, we have*

$$\max_{x':\|x-x'\|_2\leq\epsilon} D_{\mathrm{HS},\lambda}\left(\mu\left(x'\right)\|\mu(x)\right) = \Phi\left(\frac{\epsilon}{2\sigma} - \frac{\log(\lambda)\sigma}{2\epsilon}\right) - \lambda\Phi\left(-\frac{\epsilon}{2\sigma} - \frac{\log(\lambda)\sigma}{2\epsilon}\right) - \max(1-\lambda, 0) \tag{10}$$

Therefore,

$$\epsilon_\lambda = \Phi\left(\frac{\epsilon}{2\sigma} - \frac{\log(\lambda)\sigma}{2\epsilon}\right) - \lambda\Phi\left(-\frac{\epsilon}{2\sigma} - \frac{\log(\lambda)\sigma}{2\epsilon}\right) - \max(1-\lambda, 0) \tag{11}$$

Replacing the ϵ_λ of Eq.11 in Eq. 4, we have following optimisation objective:

$$\begin{aligned}
\max_{\lambda\geq 0} \quad & 1 - \lambda(1-\theta) \\
& - \left(\Phi\left(\tfrac{\epsilon}{2\sigma} - \tfrac{\log(\lambda)\sigma}{2\epsilon}\right) - \lambda\Phi\left(-\tfrac{\epsilon}{2\sigma} - \tfrac{\log(\lambda)\sigma}{2\epsilon}\right) - \max(1-\lambda, 0)\right)
\end{aligned} \tag{12}$$

To find the optimal solution, we can set the derivatives of this expression to zero with respect to λ, following the result of (Dvijotham et al. 2020), we can obtain the lower bound of $E_{x\sim q}[h(x)]$:

$$\Phi(\Phi^{-1}(\theta) - \frac{\epsilon}{\sigma}) \tag{13}$$

□

which is the same as the guarantee in (Kumar, Levine, and Feizi 2021).

E Proof of Theorem 3

Theorem 3. *(Optimisation Objective to verify l_2-norm bound by Gaussian Noise) Given the parameter λ of the Hockey-Stick divergence, we can solve the following convex optimisation problem to find the minimum bound of the mean cumulative reward:*

$$\begin{aligned}
&\max\{\eta - \underset{\tau\sim p}{\mathrm{E}}[\max((\eta + \nu\epsilon - J(\tau))\lambda, 0) + \nu\max(1-\lambda, 0)]\} \\
&s.t \quad \nu > 0, \eta \in \mathbb{R}, \eta \leq \nu(1-\epsilon) + J(\tau)
\end{aligned} \tag{14}$$

Proof. The Hockey-Stick Divergence can be represented as $f(x) = max(x-\lambda, 0) - max(1-\lambda, 0)$, which is a convex function with $f(1) = 0$, so its conjugate function is:

$$f^*_\lambda(x) = \max_{y \geq 0}(xy - max(y - \lambda, 0) + max(1 - \lambda, 0)) \tag{15}$$

Let the derivative of the expectation term with respect to y be zero, we can obtain the optimal value $y^* = \lambda$, $x - 1 \leq 0$, or $y^* = 0$ so

$$f^*_\lambda(x) = \max(x\lambda, 0) + \max(1 - \lambda, 0) \tag{16}$$

Then substitute the f^*_λ in the optimisation equation in (9), and we have the optimisation objective:

$$\begin{aligned}
&\max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu \left(\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max((\eta + \epsilon - \tfrac{J(\tau)}{\nu})\lambda, 0) + \max(1 - \lambda, 0) \right] \right) \right\} \\
&= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max\left((\nu\eta + \nu\epsilon - J(\tau))\lambda, 0 \right) + \nu \max(1 - \lambda, 0) \right] \right\} \\
&= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max((\eta + \nu\epsilon - J(\tau))\lambda, 0) + \nu \max(1 - \lambda, 0) \right] \right\}
\end{aligned}$$

with the constraint $\eta + \epsilon - \frac{J(\tau)}{\nu} \leq 1$. □

F Proof of Theorem 4

Theorem 4. *Given state $s_t \in \mathcal{S}$, suppose $p(s_t) = \mathcal{N}(s_t, \sigma^2 I_D)$, and $q(s_t) = \mathcal{N}(s'_t, \sigma^2 I_D)$, where $s'_t = \delta + s_t$. Under the constraint of $||\delta_1, \delta_2, ...||_1 \leq \epsilon$, the total variation distance for any two distributions p, q is defined as $\epsilon_{TV} = 2\Phi(\frac{\epsilon}{2\sigma^2} - 1)$, where Φ is the CDF of standard norm distribution.*

The objective function to verify perturbation bounded by l_1-norm is:

$$\begin{aligned}
&\max_{\nu>0,\eta\in\mathbb{R}} \left\{ \eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max(\eta + \nu\epsilon - J(\tau), -\tfrac{\nu}{2}) \right] \right\} \\
&s.t \quad \eta \leq \nu(\tfrac{1}{2} - \epsilon) + \min(J(\tau))
\end{aligned} \tag{17}$$

Proof. We leverage the result of theorem I in (Barsov and Ulyanov 1987), which computes the state of two Gaussian distribution $p = \mathcal{N}(\mu(x), \sigma^2 I)$ and $q = \mathcal{N}(\mu(x + \epsilon), \sigma^2 I)$ is $d_1 = \int |dq - dp| = 2(2\Phi(\epsilon/(2\sigma)) - 1)$.

As the total variance distance is $TV(q, p) = \frac{1}{2}\int |dq - dp|$, therefore, the ϵ_{TV} for the l_1-norm measure on Gaussian distribution is $2\Phi(\epsilon/(2\sigma)) - 1$.

Then we need to find the conjugate convex function expression for the total variance divergence with $f(x) = \frac{1}{2}|x - 1|$. The TV divergence can be rewritten as $f(x) = \frac{1}{2}(\max(x - 1, 0) + \max(1 - x, 0))$.

It's conjugate function can be expressed as:

$$f^*(x) = \max_{y \geq 0} \left(xy - \frac{1}{2}(\max(y - 1, 0) + \max(1 - y, 0)) \right)$$

Therefore, set the derivative to 0 with respect y, we can obtain the expression of $f^*(x)$:

$$f^*(x) = \max(x, -2/1) \tag{18}$$

with the the constraint $x \leq \frac{1}{2}$. If we substitute the f^* in the optimisation equation(9), we have the optimisation objective:

$$\begin{aligned}
&\max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu \left(\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max(\eta + \epsilon - \tfrac{J(\tau)}{\nu}, -\tfrac{1}{2}) \right] \right) \right\} \\
&= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max(\nu\eta + \nu\epsilon - J(\tau), -\tfrac{\nu}{2}) \right] \right\} \\
&= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\max(\eta + \nu\epsilon - J(\tau), -\tfrac{\nu}{2}) \right] \right\}
\end{aligned}$$

with the constraint $\eta + \nu\epsilon - J(\tau) \leq \frac{\nu}{2}$, which can be rewritten as $\eta \leq \nu(\frac{1}{2} - \epsilon) + \min(J(\tau))$ □

G Proof of Theorem 5

Theorem 5. *Given a trained smoothed policy $\tilde{\pi}$ with action space $\mathcal{A} : \{a_1, ..., a_N\}$. Suppose at each step the policy has a probability of k to select the best action a^*, and a probability of $e = 1 - k$ to select an action different from a^*. The total*

number of time steps is denoted as T. The action sequences A, selected by the smoothed policy can be viewed as on the distribution $\mu(A) = \prod_{i=1}^{T} k^{1[a_i=a_i^*]} \left(\frac{e}{N-1}\right)^{1[a_i \neq a_i^*]}$. The certification objective is: $\max_{\nu>0,\eta\in\mathbb{R}}(\eta - \mathop{\mathrm{E}}_{\tau\sim p}[\mathcal{Z}])$, where

$$\mathcal{Z} = \begin{cases} \nu + \nu^{\frac{-1}{\beta-1}}(\beta-1)\left(\frac{\max(x,0)}{\beta}\right)^{\frac{\beta}{\beta-1}} & if \quad \beta > 1 \\ -\nu + \nu^{\frac{-1}{\beta-1}}(1-\beta)\left(\frac{x}{-\beta}\right)^{\frac{\beta}{\beta-1}} & if \quad 0 \leq \beta < 1, x \leq 0 \end{cases}$$

where $x = \nu\eta + \nu\epsilon - J(\tau)$

The Rényi divergence is:

$$\left.\begin{array}{lll} \beta \geq 1: & R_\beta(q||p) = log(1 + D_f(q||p))/(\beta-1) & \text{with} \quad f(x) = x^\beta - 1 \\ \beta \in [0,1]: & R_\beta(q||p) = log(1 - D_f(q||p))/(\beta-1) & \text{with} \quad f(x) = 1 - x^\beta \end{array}\right\}$$

Therefore, if $\beta \geq 0$:

$f^*(x) = \max_{y\geq 0} xy - (y^\beta - 1)$ Take the derivative with respect to y, we have $\frac{df^*(x)}{dy} = x - \beta y^{(\beta-1)}$, and set it to zero, we have $y = (\frac{x}{\beta})^{\frac{1}{\beta-1}}$. If $x \geq 0$, $f^*(x) = 1 + (\beta-1)(\frac{x}{\beta})^{\frac{\beta}{\beta-1}}$; otherwise, $f^*(x) = 1$. It can be expressed as $1 + (\beta-1)\left(\frac{\max(x,0)}{\beta}\right)^{\frac{\beta}{\beta-1}}$. Then, substitute it in the optimisation equation (9), we can obtain the optimisation objective for $\beta \geq 1$:

$$\begin{aligned} &\max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu \left(\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[1 + (\beta-1)\left(\frac{\max(\eta+\epsilon-\frac{J(\tau)}{\nu},0)}{\beta}\right)^{\frac{\beta}{\beta-1}} \right] \right) \right\} \\ &= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \left(\nu\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[\nu + \nu^{\frac{-1}{\beta-1}}(\beta-1)\left(\frac{\max(\nu\eta+\nu\epsilon-J(\tau),0)}{\beta}\right)^{\frac{\beta}{\beta-1}} \right] \right) \right\} \end{aligned}$$

Otherwise, if $0 \leq \beta < 1$: $f^*(x) = \max_{y\geq 0} xy - (1 - y^\beta)$ Take the derivative with respect to y, we have $\frac{df^*(x)}{dy} = x + \beta y^{(\beta-1)}$, and set it to zero, we have $y = (\frac{-x}{\beta})^{\frac{1}{\beta-1}}$. If $x \leq 0$, $f^*(x) = -1 + (1-\beta)(\frac{-x}{\beta})^{\frac{-\beta}{1-\beta}}$; otherwise, $f^*(x) = \infty$. Then, substitute it in the optimisation equation (9), we can obtain the optimisation objective for $0 \leq \beta < 1$:

$$\begin{aligned} &\max_{\nu>0,\eta\in\mathbb{R}} \left\{ \nu \left(\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[-1 + (1-\beta)\left(\frac{\eta+\epsilon-\frac{J(\tau)}{\nu}}{-\beta}\right)^{\frac{\beta}{\beta-1}} \right] \right) \right\} \\ &= \max_{\nu>0,\eta\in\mathbb{R}} \left\{ \left(\nu\eta - \mathop{\mathrm{E}}_{\tau\sim p} \left[-\nu + \nu^{\frac{-1}{\beta-1}}(1-\beta)\left(\frac{\nu\eta+\nu\epsilon-J(\tau)}{-\beta}\right)^{\frac{\beta}{\beta-1}} \right] \right) \right\} \end{aligned}$$

with $\eta + \epsilon - \frac{J(\tau)}{\nu} \leq 0$.

H Related Work

Adversarial Robust of RL

Adversarial attacks on RL can be applied on inputs. Huang et al. (2017) proposed a uniform attack, which attack the RL at every time to reduce the reward. Nonetheless, this technique is characterised by its time-intensive nature. To address this concern, Lin et al. (2017) introduced a strategy-guided algorithm for attacking the agent. This algorithm triggers an attack when the discrepancy between the best and worst actions exceeds a certain threshold, indicating the efficacy of the attack. Instead of perturbing the input observation, Gleave et al. (2021) proposed to add noise in the environment. Consequently, a range of defence strategies have emerged to enhance the robustness of RL against such attacks. (Pattanaik et al. 2018; Gleave et al. 2021; Zhang et al. 2020) proposed approaches that involve utilising adversarial training, which entails incorporating augmented noise during policy training or adopting a technique where policies are simultaneously trained alongside a learned adversary by alternating optimisation.

Robust Certification

Lütjens, Everett, and How (2020) proposed to defence against the adversarial attacks targeting the observations of the agent in a certified manner tailored for DQN networks. (Fischer et al. 2019) and (Zhang et al. 2020) also discussed the robustness certification for RL, while they focused on utilising the deterministic certification adapted from (Mirman, Gehr, and Vechev 2018).Regarding the certification of cumulative rewards in reinforcement learning, the forefront contributions are represented by (Wu et al. 2021) and (Kumar, Levine, and Feizi 2021). The former, (Wu et al. 2021), concentrates on the development of a smoothed policy during testing time and the certification of per-step actions. In contrast, Kumar, Levine, and Feizi (2021) introduced an approach for certifying total rewards, obviating the need for per-step action certification, particularly when confronted with adaptive perturbations.

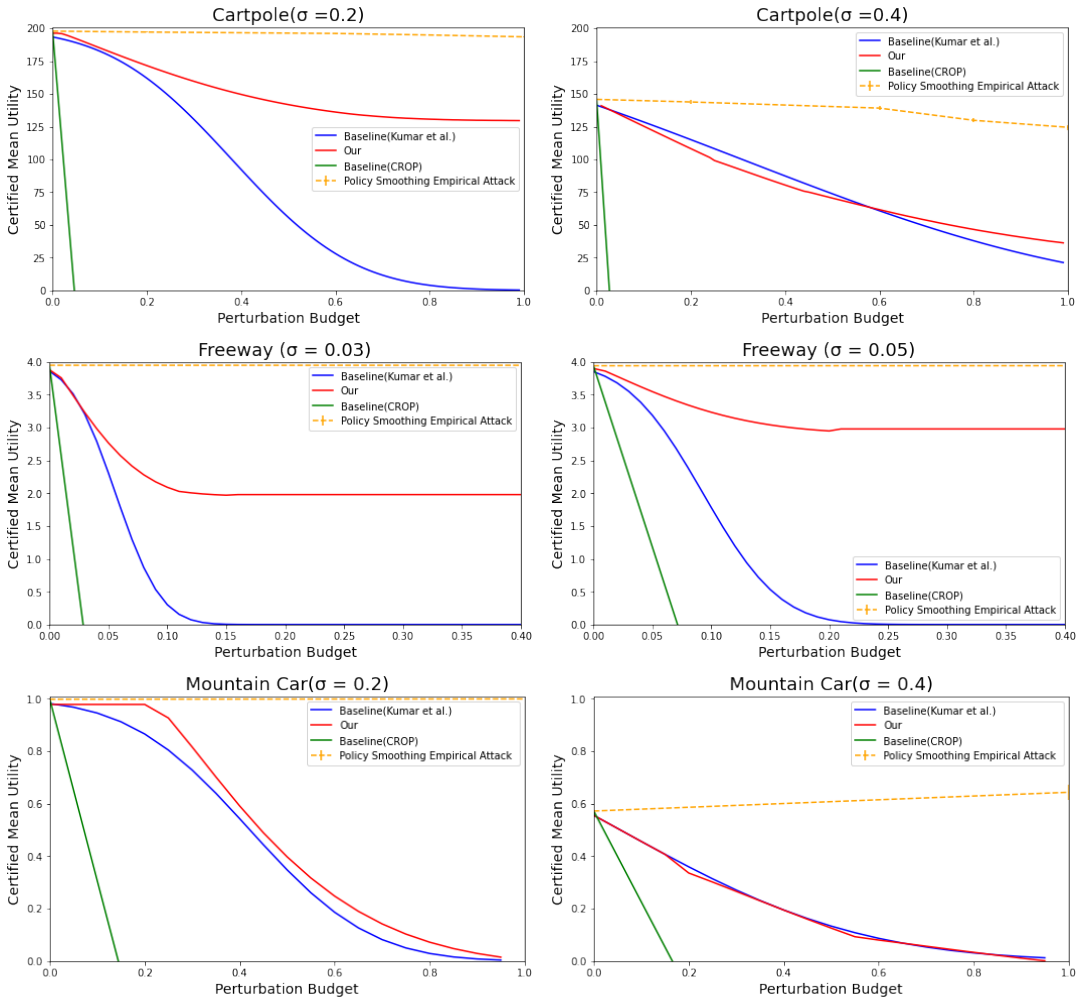

Figure 1: Certified lower bound of average cumulative reward, under l_2-norm bounded perturbation

I Extra Experimental Results and Limitations

I.1 Experiments on certifying l_2 and l_1-norm perturbation

We present the comparison results on certifying l_2-norm perturbation with (Kumar, Levine, and Feizi 2021; Wu et al. 2021) in Figure 1. The empirical attack is employed from (Kumar, Levine, and Feizi 2021), and the specific attack algorithm can be found in their Appendix M. We also implement the same training parameters as presented in the baseline. The outcomes reveal that as the smoothing factor σ rises, our approach's outcomes converge with the baseline outcomes in (Kumar, Levine, and Feizi 2021). However, it's important to note that (Wu et al. 2021) primarily emphasises certifying the resilience of per-step actions. Consequently, our results, along with those in (Kumar, Levine, and Feizi 2021), outperform the results presented in (Wu et al. 2021). Regarding the certification of l_1-norm perturbations, we adapted the empirical attack method to adhere to the l_1-norm constraint.

I.2 Certified l_0-norm perturbation

In this section, we provide additional experimental results conducted in the 'Freeway' environment, which involves three discrete actions. A comparison with the 'Cartpole' environment reveals that when the smoothing parameter p is increased, there is a noticeable decrease in the overall reward. This observation implies that augmenting the probability p of selecting alternative actions significantly undermines the efficacy of the trained policy. Notably, the limitation of our methodology lies in the fact that the 'SciPy' optimisation tool employed is based on local optimisation, thereby leading to less precise optimal outcomes. In our forthcoming research endeavours, we intend to investigate the effectiveness of global optimiser like 'DIRECT' to ascertain if they can yield more accurate optimal outcomes. Moreover, when considering the certification of l_0-norm perturbations within a continuous action space, formulating robustness criteria becomes a challenging undertaking. Unlike discrete settings, it is not

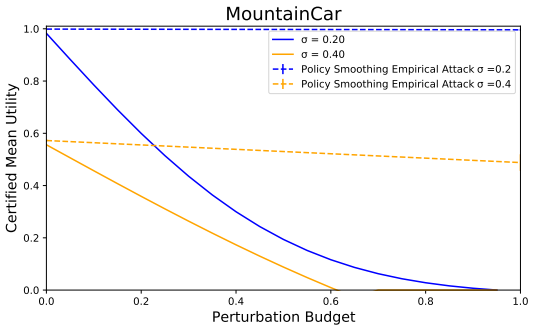

Figure 2: Certified lower bound of average cumulative reward, under l_1-norm bounded perturbation

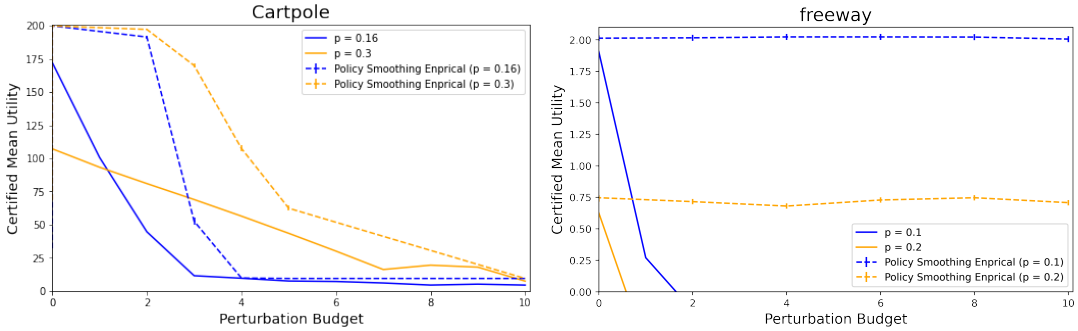

Figure 3: Certified lower bound and empirical attack result of mean cumulative reward, under l_0-norm bounded perturbation on action space in 'Cartpole' and 'Freeway' environment with discrete actions. p is the probability of changing the action.

suitable to define it straightforwardly characterised as "how many actions can be altered to attain the lower reward bound." Hence, moving forward, our future work will focus on addressing this challenge.

References

Balle, B.; and Wang, Y.-X. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, 394–403. PMLR.

Barsov, S.; and Ulyanov, V. 1987. Estimates of the proximity of Gaussian measures. *Doklady Mathematics*, 34: 462–.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. .

Dvijotham, K. D.; Hayes, J.; Balle, B.; Kolter, J. Z.; Qin, C.; György, A.; Xiao, K.; Gowal, S.; and Kohli, P. 2020. A Framework for robustness Certification of Smoothed Classifiers using F-Divergences. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2021. Adversarial Policies: Attacking Deep Reinforcement Learning. arXiv:1905.10615.

Huang, S. H.; Papernot, N.; Goodfellow, I. J.; Duan, Y.; and Abbeel, P. 2017. Adversarial Attacks on Neural Network Policies. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Kumar, A.; Levine, A.; and Feizi, S. 2021. Policy smoothing for provably robust reinforcement learning. *arXiv preprint arXiv:2106.11420*.

Kumar, A.; Levine, A.; Feizi, S.; and Goldstein, T. 2020. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems*, 33: 5165–5177.

Lin, Y.; Hong, Z.; Liao, Y.; Shih, M.; Liu, M.; and Sun, M. 2017. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 3756–3762. ijcai.org.

Lütjens, B.; Everett, M.; and How, J. P. 2020. Certified adversarial robustness for deep reinforcement learning. In *conference on Robot Learning*, 1328–1337. PMLR.

Mirman, M.; Gehr, T.; and Vechev, M. 2018. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, 3578–3586. PMLR.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In André, E.; Koenig, S.; Dastani, M.; and Sukthankar, G., eds., *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 2040–2042. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Wu, F.; Li, L.; Huang, Z.; Vorobeychik, Y.; Zhao, D.; and Li, B. 2021. Crop: Certifying robust policies for reinforcement learning through functional smoothing. *arXiv preprint arXiv:2106.09292*.

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