Certified Policy Smoothing for Cooperative Multi-Agent Reinforcement Learning

Ronghui Mu[1]*, Wenjie Ruan[2]†, Leandro Soriano Marcolino[1], Gaojie Jin[3], Qiang Ni[1]

[1]School of Computing & Communication, Lancaster University, Lancaster, LA1 4YW, UK
[2] Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK
[3] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
{r.mu, l.marcolino, n.qiang}@lancaster.ac.uk, w.ruan@exeter.ac.uk, g.jin3@liverpool.ac.uk

Abstract

Cooperative multi-agent reinforcement learning (c-MARL) is widely applied in safety-critical scenarios, thus the analysis of robustness for c-MARL models is profoundly important. However, robustness certification for c-MARLs has not yet been explored in the community. In this paper, we propose a novel certification method, which is the first work to leverage a scalable approach for c-MARLs to determine actions with guaranteed certified bounds. c-MARL certification poses two key challenges compared to single-agent systems: **(i)** the accumulated uncertainty as the number of agents increases; **(ii)** the potential lack of impact when changing the action of a single agent into a global team reward. These challenges prevent us from directly using existing algorithms. Hence, we employ the false discovery rate (FDR) controlling procedure considering the importance of each agent to certify per-state robustness. We further propose a tree-search-based algorithm to find a lower bound of the global reward under the minimal certified perturbation. As our method is general, it can also be applied in a single-agent environment. We empirically show that our certification bounds are much tighter than those of state-of-the-art RL certification solutions. We also evaluate our method on two popular c-MARL algorithms: QMIX and VDN, under two different environments, with two and four agents. The experimental results show that our method can certify the robustness of all c-MARL models in various environments. Our tool **CertifyCMARL** is available at https://github.com/TrustAI/CertifyCMARL.

1 Introduction

Recently, cooperative multi-agent reinforcement learning (c-MARL) has attracted increasing attention from researchers and is beneficial for a wide range of applications in the real world, such as autonomous cars (Shalev-Shwartz, Shammah, and Shashua 2016), traffic lights control (Van der Pol and Oliehoek 2016) and wireless communication (de Vrieze et al. 2018). As it is widely involved in safety-critical scenarios, there is an urgent need to analyze the robustness of c-MARLs.

Reinforcement learning (RL) aims to find the best actions for agents that can optimise the long-term reward by interacting with the surrounding environments (Rashid et al. 2018). When there is a team of agents, the system needs to jointly optimise each agent's actions to maximise the reward of the team. In c-MARL, as the number of agents increases, the joint action space of the agents grows exponentially, requiring the learning of policies in a decentralised manner (Oliehoek, Spaan, and Vlassis 2008). Thus, each agent learns its own policy based on its local action-observation history, and then forms a centralised action-value that is conditioned to the global state and joint actions.

Deep neural networks (DNNs) are known to be vulnerable to tiny, non-random, ideally human-invisible perturbations of the input, which can lead to incorrect predictions (Szegedy et al. 2013; Xu, Ruan, and Huang 2022; Jin et al. 2022; Wang et al. 2022; Mu et al. 2022; Yin, Ruan, and Fieldsend 2022; Ruan et al. 2019; Zhang et al. 2020b). RL has also been shown to be susceptible to perturbation in the observations of an RL agent (Huang et al. 2017; Behzadan and Munir 2017) or in environments (Gleave et al. 2019). Some adversarial defence works for RL are proposed (Donti et al. 2020; Eysenbach and Levine 2021; Shen et al. 2020; Sun et al. 2022) and then towards these defences, stronger attacks are proposed (Salman et al. 2019; Russo and Proutiere 2019). To end this repeated game, Wu et al. (2021) and Kumar, Levine, and Feizi (2021) proposed to use probabilistic approaches to provide robustness certification for RLs. Concerning c-MARL, Lin et al. (2020) addressed the challenges of attacking such systems and proposed adding perturbations to the state space. To date, the robustness certification on c-MARL has not been touched upon by the community.

Compared to the RL system with a single agent, certifying c-MARL is a more challenging task. **Challenge 1**: the action space grows exponentially with the number of agents; moreover, for each time step, the agents need to be certified simultaneously, accumulating uncertainty. **Challenge 2**: changing the action of one agent may not alter the team reward, thus, instead of following existing certification works on a single agent, new criteria should be raised to evaluate the robustness for the multi-agent system. Therefore, to cope with such challenges, we propose two novel methods to certify the robustness of each state and of the whole trajectory.

We first propose a smoothed policy where each agent chooses the most frequent action when its observation is perturbed, and then we derive the certified bound of per-

*Ronghui conducted this research when she was a visiting PhD student at the University of Exeter.

†Corresponding Author

turbation for each agent per step, within which the chosen action of the agent will not be altered. When evaluating the robustness of all agents per time step, to tackle **Challenge 1**, we identify the multiple test problem and propose to *correct* the p-value by multiplying the importance factor of each agent. We then employ the Benjamini-Hochberg (BH) procedure with corrected p-value to control the selective false discovery rate (FDR). For the certification of robustness of the global reward, to deal with **Challenge 2**, we propose a tree-search-based algorithm to find the certified lower bound of the perturbation and the lower bound of the global reward of the team under this perturbation. In this paper, we focus on certifying the robustness of value-based c-MARLs under a l_2 norm bounded attack. Our work can be easily extended to evaluate l_p norm based robustness by using different sampling distributions, such as the generalised Gaussian distribution as indicated in Hayes (2020).

Our main **contributions** can be summarised as: **i)** for the first time, we propose a solution to certify the robustness of c-MARLs, which is a *general* framework that can also be employed in a single-agent system; **ii)** we propose a new criterion to enable the *scalable* robustness certification per state for c-MARLs by considering the importance of each agent to reduce the error of selective multiple tests; and **iii)** we propose a tree-search-based method to obtain the certified lower bound of the global team reward, which enables a tighter certification bound than the state-of-the-art certification methods.

2 Background

Cooperative Multi-Agent Reinforcement Learning

Most c-MARL methods use the centralised training scheme to guide decentralised execution, such as value decomposition networks (VDN) (Sunehag et al. 2017) and QMIX (Rashid et al. 2018). In this paper, we focus on certifying the robustness of these value-based c-MARLs.

We consider a fully cooperative multi-agent game G as a Dec-POMDP (Kraemer and Banerjee 2016), which is defined by the tuple $G = \langle S, \mathcal{A}, P, r, Z, \mathcal{O}, N, \gamma \rangle$, in which each agent $n \in \{1, 2, ..., N\}$ chooses an action $a^n \in \mathcal{A}$ in each state $s \in S$ to form the joint action $\mathbf{a} = \{a^1, a^2, ..., a^N\}$. The same reward function is shared by all agents $r(s, \mathbf{a})$. γ is a discount factor. We suppose that each agent draws an observation $z^n \in Z$ given the observation function $\mathcal{O}(s, \mathbf{a})$.

Each agent has a stochastic policy $\pi^n(a^n|h^n)$ where h^n is the action-observation history $h^n \in \mathcal{H}$. The joint policy π has a joint discount return $R_t = \sum_{i=0}^{\infty}(\gamma^i r_{t+i})$ and an action-value function: $Q^\pi(s_t, \mathbf{a}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{a}_{t+1:\infty}}[R_t|s_t, \mathbf{a}_t]$. Given an action-value function Q^π, we define a greedy policy as $\pi(s_t)$: $\arg\max_{\mathbf{a}_t \in \mathcal{A}} Q^\pi(s_t, \mathbf{a}_t)$ that returns the optimal action.

Randomized Smoothing for Classification

Randomised smoothing (Cohen, Rosenfeld, and Kolter 2019) was developed to evaluate probabilistic certified robustness for classification tasks. It aims to construct a smoothed model $g(x)$, which can produce the most probable prediction of the base classifier $f(x)$ over perturbed inputs from Gaussian noise in a test instance. The smoothed classifier $g(x)$ is supposed to be provably robust to l_2-norm bounded perturbations within a certain radius:

Theorem 1. *(Cohen, Rosenfeld, and Kolter 2019) For a classifier $f : \mathbb{R} \to \mathcal{Y}$, suppose $c \in \mathcal{Y}$, let $\delta \sim \mathcal{N}(0, \sigma^2 I)$, the smoothed classifier be $g(x) := \arg\max_c \mathbb{P}(f(x + \delta) = c)$, suppose $\underline{p_A}, \overline{p_B} \in [0, 1]$, if*

$$\mathbb{P}(f(x + \delta) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x + \delta) = c), \tag{1}$$

then $g(x + \epsilon) = c_A$ for all $||\epsilon||_2 \leq R$, where

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})). \tag{2}$$

Here Φ^{-1} is the inverse cumulative distribution function (CDF) of the normal distribution.

3 Policy Smoothing for c-MARLs

In this section, we first outline an intuitive approach to certifying RL based on current classifier certification. Then, we sketch the challenges preventing the direct use of the intuitive approach and present how to address these challenges.

Problem Formulation

We aim to design a robust policy for multi-agent reinforcement learning algorithms. Following the standard setting of existing adversarial attacks on c-MARLs, e.g. (Lin et al. 2020), where the adversarial perturbation is added to each step's observation of each agent, our proposed policy is expected to be provably robust against the perturbation bounded by the l_2-norm around the observation of each agent.

Definition 1. *(Smoothed policy) Given a trained multi-agent reinforcement learning network Q^π with policy π, suppose that there are N agents, at the time step t, let $\forall s_t \in S$, given that the noise vector $\Delta_t = (\delta_t^1, ..., \delta_t^N)$ is $i.i.d$ $\mathcal{N}(0, \sigma^2 I)$, the joint smoothed policy can be represented as*

$$\tilde{\pi}(s_t) = \arg\max_{\mathbf{a}_t \in \mathcal{A}} \widetilde{Q}^\pi(s_t + \Delta_t, \mathbf{a}_t) \tag{3}$$

To certify the robustness of the smoothed policy, we define the certification robustness for a per-step action as

$$\tilde{\pi}_t(s_t) = \tilde{\pi}_t(s_t + \epsilon_t) \quad s.t. \forall \epsilon_t, ||\epsilon_t||_2 \leq D \tag{4}$$

where $\epsilon_t \in \mathbb{R}^N$ represents the maximum perturbation applied to the observations of each agent at the t-th time step. In other words, for each agent, in the presence of the l_2-norm bounded perturbation in each state, the smoothed policy is expected to return the same action that is most likely to be selected in the unperturbed state s_t.

Intuitive Approach

Intuitively, the randomised smoothing can be adapted to certify the robustness of the per-state action in RLs by replacing the classifier $f(x)$ with policy $\pi(s_t)$. For the certification of

Algorithm 1: Intuitive Policy Smoothing for Certifying Per-state Action

Input: Trained Q^π with N agents
Parameter: sampling times M; Gaussian distribution parameter σ; confidence parameter α

1: **function** SMOOTHING(M, Q, α, σ)
2: **for** $m \leftarrow 1, M$ **do** ▷ Get smoothed policy $\tilde{\pi}$
3: generate $\Delta_m = (\delta_m^1, ..., \delta_m^N)$ i.i.d $\mathcal{N}(0, \sigma^2 I)$
4: $s' \leftarrow s + \Delta_m$
5: $\mathbf{a} \leftarrow \pi(s')$
6: Add $\mathbf{a} \rightarrow Actlist$
7: **return** $Actlist$
8: $Actlist \leftarrow$ SMOOTHING$(M, Q^\pi, \alpha, \sigma)$
9: $\mathbf{a}^m, \mathbf{a}^r, ct_1, ct_2 \leftarrow$ Top two action sets with their counts
10: **if** $BioPVALUE(ct_1, ct_1 + ct_2, 0.5) \leq \alpha$ **then**
11: $Cert \leftarrow True$ ▷ Get certified radius for $\tilde{\pi}$
12: $\underline{p_{\mathbf{a}^m}}, \overline{p_{\mathbf{a}^r}} \leftarrow MultiConBnd(Counts(Actlist), \alpha)$
13: $D \leftarrow \frac{\sigma}{2}\left(\Phi^{-1}\left(\underline{p_{\mathbf{a}^m}}\right) - \Phi^{-1}\left(\overline{p_{\mathbf{a}^r}}\right)\right)$
14: **else**
15: $Cert \leftarrow False, D \leftarrow 0$
16: **return** $\mathbf{a}, d, Cert$

each step, Monte Carlo randomised sampling is used to estimate the smoothed policy $\tilde{\pi}$. As shown in Algorithm 1, we record the action vector $\mathbf{a}$, which is a combination of actions taken by all agents at each sampling step. The most likely selected action set is chosen as the action in $\tilde{\pi}$. A larger number of samples can be used to estimate the lower bound on the probability ($\underline{p_{\mathbf{a}^m}}$) of the most frequently selected action set, $\mathbf{a}^m$, and the upper bound on the probability ($\overline{p_{\mathbf{a}^r}}$) of the second most frequently selected ("runner-up") action, $\mathbf{a}^r$. The function MULTICONBND in Algorithm 1 is based on a Chi-Square approximation (Goodman 1965), which takes the number of observations for each category as input and returns the $(1-\alpha)$ confidence levels.

Proposition 1. *If the certification in Algorithm 1 returns the action set* $\mathbf{a}^m : (a^{m,1}, a^{m,2}, ..., a^{m,N})$ *in the time step t with a certified radius* $D = \frac{\sigma}{2}\left(\Phi^{-1}\left(\underline{p_{\mathbf{a}^m}}\right) - \Phi^{-1}\left(\overline{p_{\mathbf{a}^r}}\right)\right)$ *then with probability at least* $(1-\alpha)$*, the smoothed policy* $\tilde{\pi}(s_t + \epsilon_t)$ *chooses the action* $\mathbf{a}^m$*,* $\forall ||\epsilon_t||_2 \leq D$.

Proof is provided in **Appendix**[1] **A**. The intuition behind the method shown in Algorithm 1 is similar to the certification procedure for classification through randomised smoothing (Cohen, Rosenfeld, and Kolter 2019). The BIOPVALUE is applied to calculate the p-value of the two-sided hypothesis test to choose the action $\mathbf{a}^m$. However, rather than abstaining from the action when the p-value does not meet the confidence level, we set the certified radius of this step as $D = 0$ to indicate that the certification failed, since the RL relies on decisions of multiple steps. When $n = 1$, the algorithm can be used to certify RLs with a single agent as Wu et al. (2021), but instead of using the smoothed action value function Q^π, we utilise the frequency of occurrence of each action to determine which action to be selected. Since c-MARLs are trained under the premise that each agent would always select the best action, they do not reliably anticipate the team reward when some agents behave badly.

In the c-MARLs, there are some additional challenges that preclude us from using this intuitive certification criterion.

Challenge 1. The perturbation D added to the observation of each agent can be different. For c-MARLs, each agent develops its own policy to choose its action. If the certified bound is calculated using Algorithm 1, all agents will engage with the same perturbation bound, making the results less accurate for each agent. As one agent can be more robust than the other, the same perturbation added to the agents will lead to different performances, which provides the need to certify the robustness of each agent. Thus, we will first consider certifying the robustness for every agent and then estimating the robustness in each state for all agents. To reduce the computation cost, we can sample from the joint policy $\pi(s')$ instead of each agent's policy separately. To this end, we can change $\mathbf{a}^m, \mathbf{a}^r$ in Algorithm 1 (Line 9) to the two most likely actions $(a^{m,n}, a^{r,n})$ for each agent and then calculate the corresponding lower bound $\underline{p_{a^{m,n}}}$ on probability $\mathbb{P}(\tilde{\pi}^n(z^n) := a^{m,n})$ and upper bound $\overline{p_{a^{r,n}}}$ for choosing the "runner up" action, $a^{r,n}$. The certified bound for each agent per state can be computed as:

Corollary 1. *(Certification for the actions of each agent in each state) In state s, given the joint smoothed policy* $\tilde{\pi}(s) = \{\tilde{\pi}^1(z^1), ..., \tilde{\pi}^N(z^N)\}$*, we can obtain the certified bound in state s for each agent to guarantee* $\tilde{\pi}^n(z^n + \epsilon) := a^{m,n}, \forall ||\epsilon||_2 \leq d_n$*:*

$$d_n = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_{a^{m,n}}}) - \Phi^{-1}(\overline{p_{a^{r,n}}})) \tag{5}$$

Proof is provided in **Appendix A**. Finally, the most likely chosen action for each agent can be combined as the final action set $\{a^{m,1}, a^{m,2}, ..., a^{m,N}\}$ and the certified bound at each step can be defined as:

Definition 2. *Given the certified bound obtained for each agent in state s,* $\{d_1, d_2, ..., d_N\}$*, the certified bound in this state for all agents is determined by the least robust agent:* $D = min\{d_1, d_2, ..., d_N\}$.

Challenge 2. If we choose the bound of the least robust agent as the bound for all agents per state, the confidence level decays. As Proposition 1 indicates, on each call of certification, the certified robustness bound obtained only holds with confidence level $(1-\alpha)$. As we sample noise from the Gaussian distribution *independently*, the hypothesis tests are *independent*. Based on Definition 2, to calculate the certified bound for each state, we have the following constraint for the probability of making an error:

$$\mathbb{P}(\bigvee_{n \in N} n\text{-th agent's cert failed}) \leq \min\left(\sum_n \mathbb{P}(n\text{-th agent's cert failed}), 1\right) = \min(N\alpha, 1)$$

Therefore, for multiple tests, without any control on the error, the probability of making an error will increase with the number of tests. Suppose that there are T steps in the entire trajectory, we will have $N * T$ tests in total, which can be a

[1]All appendixes of this paper can be found at https://github.com/TrustAI/CertifyCMARL/blob/main/appendix.pdf

great challenge. To address this problem, for certifying per-state actions, the confidence level can be reduced to α/N. Additionally, we can first perform agent selection to control the selective error by considering the importance of each agent, since sometimes an agent changing its action will not diminish the team reward. Moreover, to evaluate the global certification bound, we propose a tree-search-based method to find the lower bound of the team reward. In Section 4 and 5, we will detail our proposal to certify the robustness of per-state actions and global reward.

4 Robustness Certification for Per-State Action with Correction

Multiple Hypothesis Testing

Corollary 2. *(Certified bound per state) In state s, given N agents with action $\mathbf{a}$, the joint policy is $\pi(s) = \{\pi^1(z^1), ..., \pi^N(z^N)\}$. Suppose that the observation of each agent is perturbed by random noise δ^n, where $\delta^n \sim \mathcal{N}(0, \sigma^2 I)$. $\forall n \in N$, if $\mathbb{P}(\pi^n(z^n + \delta^n) := a^{m,n}) \geq 0.5$, we can compute the certified bound by **Definition 2**.*

The proof is presented in **Appendix B**. To obtain the certified bound for all agents per state, we can employ Corollary 2, and, as suggested, for each agent, we need to ensure that condition $\mathbb{P}(\tilde{\pi}^n(z^n) := a^{m,n}) \geq 0.5$ is satisfied. Hence, after sampling, with the count (ct_1^n), we can implement the one-sided binomial test to obtain its p-value pv_n. These p-values can be processed to indicate which tests should be accepted under $(1-\alpha)$ confidence.

Definition 3. *(Hypothesis Test) The hypothesis test with null hypothesis for each agent is $H_0 : \mathbb{P}(\tilde{\pi}^n(z^n) := a^{m,n}) < 0.5$, and the alternative is $H_1 : \mathbb{P}(\tilde{\pi}^n(z^n) := a^{m,n}) \geq 0.5$*

In the hypothesis test, if the null hypothesis H_0 is true, we can determine the p-value, which is the probability of finding a statistic that is equally extreme as the observed one or more extremes. Given the statistical test in **Definition 3**, if the p-value is below the confidence level, we can reject the null hypothesis, which means that the bound is certified.

In multiple hypothesis tests, the probability of the occurrence of false positives (FP) will increase, where the FP denotes that we reject the null hypothesis when it is true, which is also called *type I error*. Suppose that the confidence level is α, and the probability of FP is expected to be less than α. To control *type I error* for multiple tests with H tests, the family-wise error rate (FWER) is introduced, which changes α for each test to α/H. However, it is still conservative, which can increase the true negative rate (i.e., *type II error*).

To solve this problem, Benjamini and Hochberg (1995) proposed the false discovery rate (FDR) to find the expected false positive portion. The FDR method applies a corrected p-value for each test case, achieving a better result: testing for as many positive results as possible while keeping the false discovery rate within an acceptable range. The Benjamini-Hochberg (BH) procedure first sorts the p-values of tests in ascending order and then finds the largest k such that $p_k \leq k\alpha/H$, rejecting null if the p-value is below p_k. Fithian, Sun, and Taylor (2014) then proposed selective hypothesis tests by applying inference to the selected model to

Algorithm 2: Certified Robustness Bound of the Perturbation for Actions of Each State with Correction (CRSC)

Input: Trained Q^π; N agents;
Parameter: sampling size M; Gaussian distribution parameter σ; confidence parameter α

1: $Actlists \leftarrow \text{SMOOTHING}(M, Q^\pi, \alpha, \sigma)$
2: $a^{m,n}, a^{r,n}, ct_1^n, ct_2^n \leftarrow Counts(Actlists[n])$ for $n \in N$
3: $IF \leftarrow IF\_function(Q^\pi, Actlists)$ ▷ Obtain importance factor for agent
4: $pv_n \leftarrow BioPVALUE(ct_1^n, M, 0.5)$ for $n \in N$
5: $c_n \leftarrow BHproc((pv_n * IF[n]), \alpha)$ for $n \in N$
6: If $\neg c_n : d_n \leftarrow 0$ ▷ Remove failed agent
7: $\mathcal{I}_{\boldsymbol{cert}} := \{n \mid d_n \neq 0\}$ ▷ Obtain certified agent set
8: Compute d_n for each agent in $\mathcal{I}_{\boldsymbol{cert}}$
9: $D = min(d_n | n \in \mathcal{I}_{\boldsymbol{cert}})$
10: **return** D, $\mathcal{I}_{\boldsymbol{cert}}$

control the selective *type I error*, which controls the global error as $\frac{\mathbb{E}[\#FalseRejections]}{\mathbb{E}[\#H_0 Selected]} \leq \alpha$. Inspired by the selective hypothesis tests, we propose to multiply every agent's importance factor with its p-value to control the selective FDR via executing the BH procedure on the corrected p-values.

Measuring the Importance of Agents

To obtain each agent's important factor, we can measure each agent's contribution to the team reward at each state. We adopt the advantage function proposed in COMA (Foerster et al. 2018), which is used to decentralise agents by estimating the individual reward during training. As the importance factor defined in **Definition** 4, it is applied to examine the behaviour of the current action of the agent.

Definition 4. *For each agent n, the importance factor IF^n of each agent is computed by comparing the Q value of the current action a^n with the counterfactual reward baseline, which is obtained by altering the action of agent n, $a^{n'}$, and keeping the other agents' actions $\mathbf{a}^{-n}$ unchanged:*

$$IF^n(s, \mathbf{a}) = Q(s, \mathbf{a}) - \sum_{a^{n'} \in \mathcal{A}} \mathbb{P}(\tilde{\pi}^n(s) := a^{n'}) \cdot Q\left(s, \left(\mathbf{a}^{-n}, a^{n'}\right)\right)$$

Algorithm 2 shows the process for certifying the robustness of the actions of each state while controlling the error. To correct the p-value in the multiple tests, we adapt the p-value for each test by multiplying it with the agent's importance factor (Line 4). Then we can perform the BH procedure (Line 5) to determine which tests should be rejected. Lastly, we obtain the set of certified agents $\mathcal{I}_{\boldsymbol{cert}}$.

Theorem 2. *For each agent in $\mathcal{I}_{\boldsymbol{cert}} := \{n \mid d_n \neq 0\}$, the action can be certified as $\tilde{\pi}^n(z^n + \epsilon^n) = \tilde{\pi}^n(z^n)$, where $\|\epsilon^n\|_2 \leq D := min(d_n), \forall n \in \mathcal{I}_{\boldsymbol{cert}}$.*

Proof. Considering each agent independently, given that agent n updates its policy $\tilde{\pi}^n(z^n)$ in each state, under the condition $\mathbb{P}(\tilde{\pi}^n(z^n) := a^{m,n}) > 0.5$, we can obtain the lower bound probability of selecting the $a^{m,n}$ and the upper-bound probability for the "runner-up" action, $a^{r,n}$, for each

Algorithm 3: Tree-Search-based certified robustness bound and global reward (T-CRGR)

Input: Trained Q^{π}; N agents; confidence parameter α
Parameter: sampling times M; Gaussian distribution parameter σ

function GETNODE(s)
  $Actlists \leftarrow$ SMOOTHING$(M, Q^{\pi}, \alpha, \sigma)$
  $IF \leftarrow IF\_function(Q, Actlists)$
  $A\_dic, d\_list \leftarrow \emptyset$
  for $n \in \mathcal{I}_{\boldsymbol{agent}}$ **do**
    $a^{m,n}, a^{r,n}, ct_1^n, ct_2^n \leftarrow Counts(Actlists)$
    $pv_n \leftarrow BioPVALUE(ct_1^n, M, 0.5)$
    if $pv_n * IF[n] > \alpha$ **then**
      $A\_dic[n] \leftarrow A\_dic[n] \cup \{a^{m,n}, a^{r,n}\}$
      $\underline{p_1} \leftarrow BioConBnd(ct_1^n + ct_2^n, M, 1-\alpha)$
    else
      $A\_dic[n] \leftarrow A\_dic[n] \cup \{a^{m,n}\}$
      $\underline{p_1} \leftarrow BioConBnd(ct_1^n, M, 1-\alpha)$
    $d\_list \leftarrow d\_list \cup (\sigma\Phi^{-1}(\underline{p_1}))$
  $d \leftarrow min(d\_list)$
  return A_dic, d
function SEARCH($d, s, \mathbf{a}, R, done$):
  if $R \geq R_{min}$ **then** ▷ Prune the tree
    return 0
  if $done$ **then**
    $R_{min} \leftarrow min(R, R_{min})$
    return 0
  $A\_dic, d_{new} \leftarrow$ GETNODE(s)
  $d \leftarrow min(d_{new}, d), Action\_list \leftarrow A\_dic$
  for $\mathbf{a}$ in $Action\_list$ **do**
    $s_{new}, done \leftarrow env.step(\mathbf{a}, s)$
    SEARCH$(d, s_{new}, \mathbf{a}, R + Q(s, \mathbf{a}), done)$

agent and then compute the certified bound d_n. The minimum certified bound holds for any agent that satisfies the condition, denoted by the set $\mathcal{I}_{\boldsymbol{cert}}$. □

5 Robustness Guarantee on Global Reward

To certify the bound of global reward under the certified perturbation bound for each step, the CRSC is no longer applicable, as it cannot find the lower bound of global reward. Therefore, we propose a tree-search-based method to find the global lower bound of the team reward.

The insight of implementing the search tree is that, if we cannot certify the bound of perturbation at some time steps for some agent, we can take the second most frequent action, which will result in a new trajectory. Then we can explore the new trajectory by developing it as an expanded branch of the search tree, which may result in a lower global reward. Thus, the minimum reward can be determined as the certified lower bound of the global reward after exploring all trajectories. The main function is presented in Algorithm 3. As it shows, at first, we figure out all possible actions to formulate the action list to be explored using the function GETNODE. Then we perform the SEARCH function to expand the tree based on each action node. Once all new trajectories have been explored, we obtain the certified bound of perturbation and the minimum reward among all leaf nodes. We also apply pruning to control the size of the search tree, which requires the reward in the environments to be non-negative. When the cumulative reward of the current node has already reached the lower bound, it can be pruned, as the subsequent tree will not lead to a lower bound.

6 Experiments

Baseline In single-agent environments, we compare our method with CROP-LORE (Wu et al. 2021). We follow the same setting as CROP-LORE for a fair comparison. For certifying the c-MARLs, since there is no existing solution, we apply the PGD attack (Kurakin, Goodfellow, and Bengio 2018) to demonstrate the validity of the certified bounds.

Environments For the single agent environment, we use the "Freeway" in OpenAI Gym (Brockman et al. 2016) For the case of c-MARLs, we choose two environments "Checkers" with *two* agents and "Switches" with *four* agents from ma-gym (Koul 2019). Details of the environments are given in **Appendix D**. Extra experiments in "Traffic Junction" with four and ten agents are given in **Appendix E**.

RL Algorithms We apply our method to certify the DQN trained by SA-MDP (PGD) and SA-MDP (CVX) (Zhang et al. 2020a) in the single-agent setting. For c-MARLs, we use VDN (Sunehag et al. 2017) and QMIX (Rashid et al. 2018), which are well-established value-based algorithms.

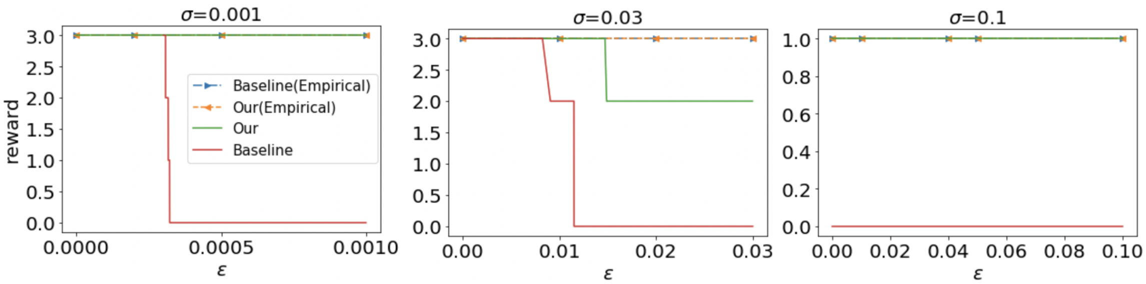
Experiments setup For all experiments, we sample noise 10,000 times for smoothing and set the discount factor γ to 1.0. In the single-agent environment, we follow the same setting as the baseline, where the time step is 200 and the confidence level is $\alpha = 0.05$. For c-MARLs, $\alpha = 0.01$.

Evaluate the Robustness of the Global Reward

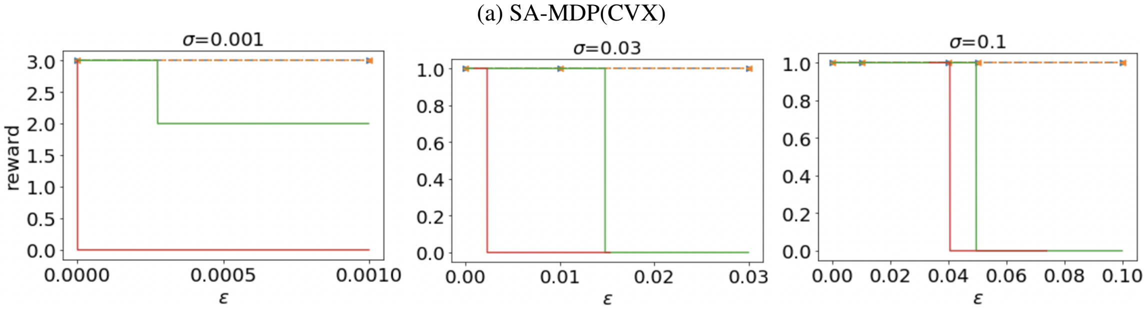
Compared with baseline on single agent The baseline develops the smoothed policy based on the action-value function bounded by Lipschitz continuous, while our method is based on the probability of selecting the most frequent action. To make a fair comparison, we employ the same search tree structure as the baseline, which organises all possible trajectories and grows them by increasing the certified bound to choose an alternative action. The technical details are given in **Appendix C**.

As shown in Figure 1, our method obtains a tighter bound than the baseline. Since we measure the probability of selecting actions instead of the action value function to calculate the bound and choose an action, we do not include the actions that have never been chosen in the possible action list, leading to a more reasonable action selection mechanism, resulting in a tighter calculated bound. Moreover, the Lipschitz continuity is used to compute the upper bound of the smoothed value function in the baseline, which is less tight than our bound based on high-probability guarantees.

Lower bound of global reward for c-MARLs In Table 1, we show the results of the lower bound of global reward under the minimum certified bound of perturbation ϵ_{cert}. To perform pruning, the per-step reward in each environment is set to be non-negative. However, as the global reward obtained for QMIX on Switch are below zero, for this case, we



(a) SA-MDP(CVX)



(b) SA-MDP(PGD)

Figure 1: Comparing the robustness certification of the total reward for SA-MDP in Freeway with Wu et al. (2021). Solid lines are the certified lower bounds of reward, and dashed lines indicate the empirical results under PGD attack.

Models	Game	No.agent	$\sigma = 0.03$			$\sigma = 0.06$			$\sigma = 0.1$		
			ϵ	Reward		ϵ	Reward		ϵ	Reward	
				Our	PGD		Our	PGD		Our	PGD
VDN	Checkers	2	0.0117	79.84	79.84	0.0221	79.84	79.84	0.0309	79.84	79.84
QMIX	Checkers	2	0.0144	19.96	19.96	0.0369	19.96	19.96	0.0384	19.96	19.96
VDN	Switch	4	0.0147	19.4	19.4	0.284	14.4	19.4	0.036	14	14.4
QMIX *	Switch	4	0.0173	-20	-20	0.0233	-20	-20	0.038	-20	-20

Table 1: Lower bound of global reward under the minimum certified bound of perturbation ϵ, where the line with '*' denotes that we run the trajectory to the end without pruning to obtain the certified reward.

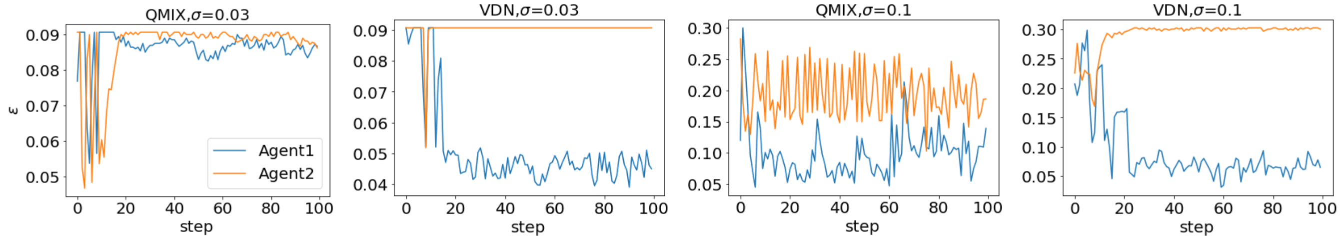
run each trajectory to the end without pruning to calculate the global reward. We can see that VDN obtains a higher reward compared to QMIX but is less robust (has lower ϵ_{cert}). This is because, during the training process, VDN simply adds rewards obtained by the two agents to achieve a centralisation, leading one agent to choosing a simpler strategy once another agent has learnt a useful strategy. On the other hand, QMIX employs a more complex network to centralise the agents instead of only adding their rewards, which helps the network to capture more complex interrelationships between different agents and encourage each one to learn. This leads to VDN achieving higher rewards faster than QMIX, but being more vulnerable to perturbations.

Evaluate the Robustness for Each State

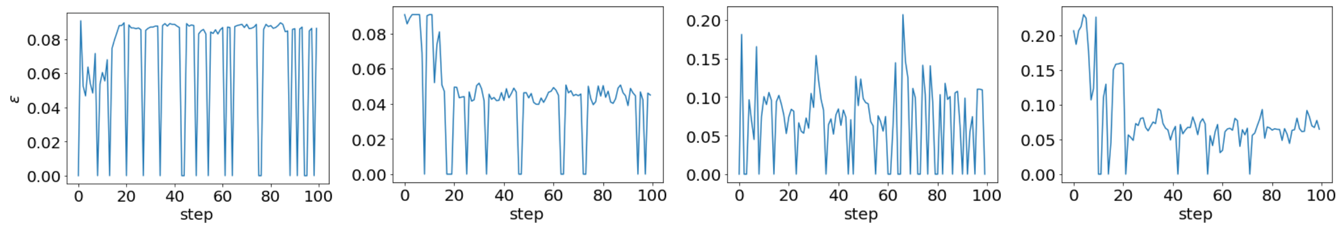
In Figure 2, we present the certified perturbation bounded by l_2 norm for each agent and for all agents at each state separately. We see that in Checkers with two agents, the certified bound for each agent (trained by QMIX) is close to each other when the smoothing variance σ is 0.03. When we increase the variance to 0.1, Agent2 engages a slightly higher bound than Agent1, which means that Agent2 is more robust. For the agents trained by VDN, Agent2 always has a much higher robustness bound than Agent1. It may be because when training QMIX, all agents are expected to learn useful strategies, while VDN only needs some agents to learn well, and others may use lazier strategies, which results in a big divergence in the robustness between agents of VDN. In Switch with four agents, we observe that, by applying our p-value corrected method, the locally certified bound at each step will not always take the minimum bound among all agents and ignore the bound of agents with low impact.

7 Related work

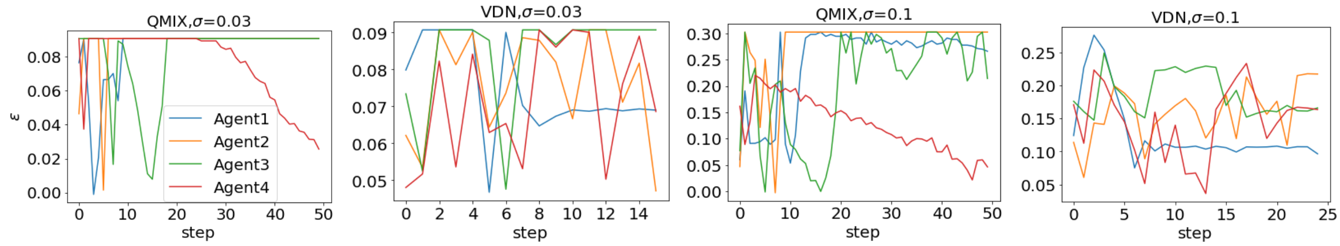
Adversarial Attacks on DRLs Existing attack solutions mainly focused on attacking single-agent RL systems, such as Huang et al. (2017); Lin et al. (2017); Kos and Song (2017); Weng et al. (2019). For attacking c-MARLs, there are notably two existing works. Lin et al. (2020) proposed to train a policy network to find a wrong action that the victim



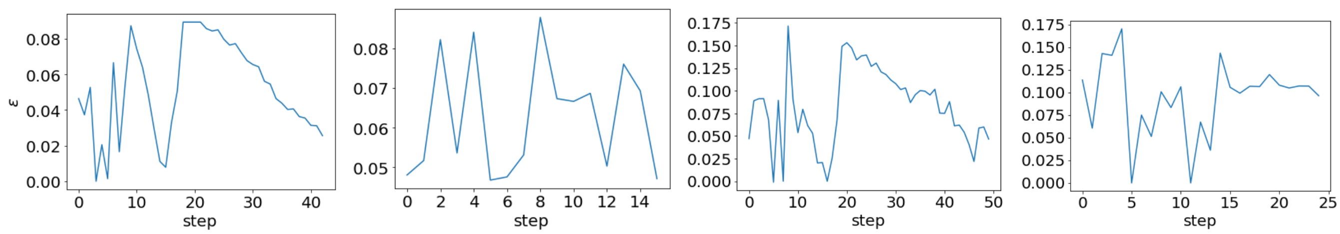
(a) Certified bound of perturbation for actions of each state of each agent in Checkers



(b) Certified bound of perturbation for actions of each state in Checkers



(c) Certified bound of perturbation for per-state action of each agent in Switch.



(d) Certified bound of perturbation for actions of each state in Switch

Figure 2: Certified robustness for per-step action.

agent is expected to take and set it as the targeted adversarial example. Pham et al. (2022) then proposed to craft a stronger adversary by using a model-based approach.

Robustness Certification of DRLs Majority research on robustness certification concentrated on DNNs (Wang et al. 2023; Zhang, Ruan, and Xu 2023; Ruan, Huang, and Kwiatkowska 2018; Wu et al. 2020; Zhang, Ruan, and Fieldsend 2022; Wang and Ruan 2022). Certification on DRLs is still in its infancy. Lütjens, Everett, and How (2020) first proposed a certified defence on the observations of DRLs. Zhang et al. (2020a) then provided empirically provable certificates to ensure that the action does not change at each state. However, this method cannot certify the robustness of the reward if the action is changed under attacks. To tackle this problem, Kumar, Levine, and Feizi (2021) proposed to directly certify the total reward via randomised smoothing-based defence, but it cannot certify the robustness at action level. Recently, Wu et al. (2021) proposed a policy smoothing method based on the randomised smoothing of the action-value function. However, all existing methods can only work on single-agent systems. To the best of our knowledge, this paper is the first work to certify the robustness of cooperative multi-agent RL systems.

8 Conclusion

We propose the first robustness certification solution for c-MARLs. By combining the FDR-controlling strategy with the importance factor of each agent, we certify the actions for each state while mitigating the multiple testing problem. In addition, a tree-search-based algorithm is applied to obtain a lower bound of the global reward. Our method is also applicable to single-agent RL systems, where it can obtain tighter bound than the state-of-the-art certification methods.

Acknowledgements

This work is supported by the Partnership Resource Fund of ORCA Hub via the UK EPSRC under project [EP/R026173/1]. Ronghui was funded by the Faculty of Science and Technology at Lancaster University. We thank the High-End Computing facility at Lancaster University for the computing resources. We also thank Matheus Alves and Peng Gao for proofreading.

References

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

Benjamini, Y.; and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1): 289–300.

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

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 1310–1320. PMLR.

de Vrieze, C.; Barratt, S.; Tsai, D.; and Sahai, A. 2018. Cooperative multi-agent reinforcement learning for low-level wireless communication. *arXiv preprint arXiv:1801.04541*.

Donti, P. L.; Roderick, M.; Fazlyab, M.; and Kolter, J. Z. 2020. Enforcing robust control guarantees within neural network policies. *arXiv preprint arXiv:2011.08105*.

Eysenbach, B.; and Levine, S. 2021. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*.

Fithian, W.; Sun, D.; and Taylor, J. 2014. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2019. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*.

Goodman, L. A. 1965. On simultaneous confidence intervals for multinomial proportions. *Technometrics*, 7(2): 247–254.

Hayes, J. 2020. Extensions and limitations of randomized smoothing for robustness guarantees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 786–787.

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

Jin, G.; Yi, X.; Huang, W.; Schewe, S.; and Huang, X. 2022. Enhancing Adversarial Training with Second-Order Statistics of Weights. *CVPR*.

Kos, J.; and Song, D. 2017. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*.

Koul, A. 2019. ma-gym: Collection of multi-agent environments. https://github.com/koulanurag/ma-gym. Accessed: 2022-9-10.

Kraemer, L.; and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190: 82–94.

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

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, 99–112. Chapman and Hall/CRC.

Lin, J.; Dzeparoska, K.; Zhang, S. Q.; Leon-Garcia, A.; and Papernot, N. 2020. On the robustness of cooperative multi-agent reinforcement learning. In *2020 IEEE Security and Privacy Workshops (SPW)*, 62–68. IEEE.

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

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

Mu, R.; Ruan, W.; Marcolino, L. S.; and Ni, Q. 2022. 3DVerifier: efficient robustness verification for 3D point cloud models. *Machine Learning*, 1–28.

Oliehoek, F. A.; Spaan, M. T.; and Vlassis, N. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32: 289–353.

Pham, N. H.; Nguyen, L. M.; Chen, J.; Lam, H. T.; Das, S.; and Weng, T.-W. 2022. Evaluating Robustness of Cooperative MARL: A Model-based Approach. *arXiv preprint arXiv:2202.03558*.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4295–4304. PMLR.

Ruan, W.; Huang, X.; and Kwiatkowska, M. 2018. Reachability analysis of deep neural networks with provable guarantees. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, 2651–2659.

Ruan, W.; Wu, M.; Sun, Y.; Huang, X.; Kroening, D.; and Kwiatkowska, M. 2019. Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the Hamming Distance. In *The 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. IJCAI.

Russo, A.; and Proutiere, A. 2019. Optimal attacks on reinforcement learning policies. *arXiv preprint arXiv:1907.13548*.

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

Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Shen, Q.; Li, Y.; Jiang, H.; Wang, Z.; and Zhao, T. 2020. Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*, 8707–8718. PMLR.

Sun, Y.; Zheng, R.; Hassanzadeh, P.; Liang, Y.; Feizi, S.; Ganesh, S.; and Huang, F. 2022. Certifiably Robust Policy Learning against Adversarial Communication in Multi-agent Systems. *arXiv preprint arXiv:2206.10158*.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

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

Van der Pol, E.; and Oliehoek, F. A. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)*, 1.

Wang, F.; Xu, P.; Ruan, W.; and Huang, X. 2023. Towards Verifying the Geometric Robustness of Large-scale Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*.

Wang, F.; Zhang, C.; Xu, P.; and Ruan, W. 2022. Deep learning and its adversarial robustness: A brief introduction. In *HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation*, 547–584.

Wang, Z.; and Ruan, W. 2022. Understanding Adversarial Robustness of Vision Transformers via Cauchy Problem. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'22)*.

Weng, T.-W.; Dvijotham, K. D.; Uesato, J.; Xiao, K.; Gowal, S.; Stanforth, R.; and Kohli, P. 2019. Toward evaluating robustness of deep reinforcement learning with continuous control. In *International Conference on Learning Representations*.

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*.

Wu, M.; Wicker, M.; Ruan, W.; Huang, X.; and Kwiatkowska, M. 2020. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807: 298–329.

Xu, P.; Ruan, W.; and Huang, X. 2022. Quantifying safety risks of deep neural networks. *Complex & Intelligent Systems*, 1–18.

Yin, X.; Ruan, W.; and Fieldsend, J. 2022. DIMBA: discretely masked black-box attack in single object tracking. *Machine Learning*, 1–19.

Zhang, C.; Ruan, W.; and Xu, P. 2023. Reachability Analysis of Neural Network Control Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Boning, D.; and Hsieh, C.-J. 2020a. Robust deep reinforcement learning against adversarial perturbations on observations. *arXiv preprint arXiv:2003.08938*.

Zhang, T.; Ruan, W.; and Fieldsend, J. E. 2022. PRoA: A Probabilistic Robustness Assessment against Functional Perturbations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'22)*.

Zhang, Y.; Ruan, W.; Wang, F.; and Huang, X. 2020b. Generalizing universal adversarial attacks beyond additive perturbations. In *2020 IEEE International Conference on Data Mining (ICDM'20)*, 1412–1417. IEEE.