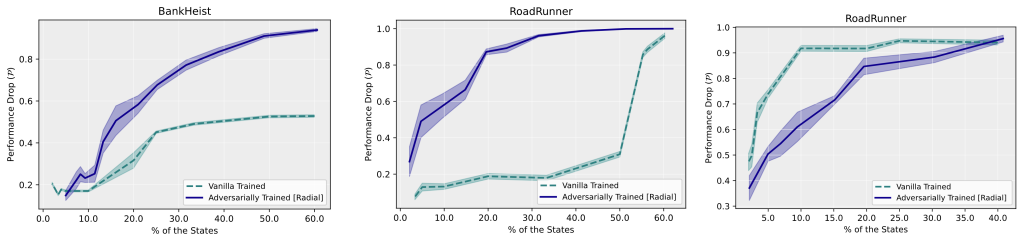
EXPOSING THE SILENT HIDDEN IMPACT OF CERTIFIED TRAINING IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

1 SUPPLEMENTARY MATERIAL

1.1 OVERESTIMATION, INACCURACIES AND INCONSISTENCIES IN ADVERSARIAL TRAINING: RADIAL



Performance drop $\mathcal{P}_2(p)$ Performance drop $\mathcal{P}_2(p)$ Performance drop $\mathcal{P}_w(p)$

Figure 1: Left: Performance drop $\mathcal{P}_2(p)$ with respect to action modification a_2 for RADIAL adversarially trained deep neural policies Oikarinen et al. (2021) and vanilla trained policies for BankHeist. Center: Performance drop $\mathcal{P}_2(p)$ with respect to action modification a_2 for RADIAL adversarially trained deep neural policies Oikarinen et al. (2021) and vanilla trained policies for RoadRunner. Right: Performance drop $\mathcal{P}_w(p)$ with respect to action modification a_w for the RADIAL adversarially trained deep neural policy and the vanilla trained deep neural policy.

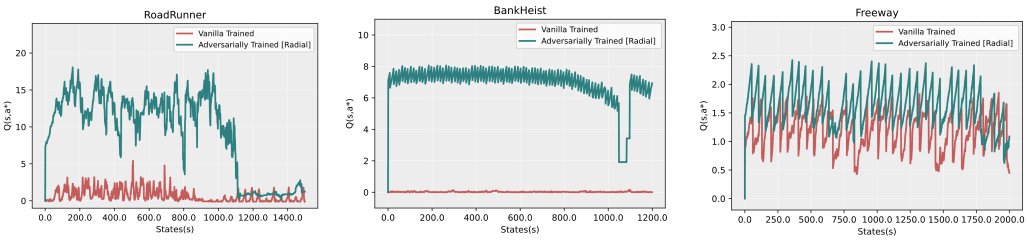


Figure 2: Q-value of the best action a^* over the states for the RADIAL adversarially trained deep neural policy proposed by Oikarinen et al. (2021) and vanilla trained deep neural policy.

The left and center column of Figure 1 demonstrate the performance drop $\mathcal{P}_2(p)$ with respect to action modification a_2 for the RADIAL adversarially trained deep reinforcement learning policy proposed by Oikarinen et al. (2021) and the vanilla trained deep reinforcement learning policy in BankHeist and RoadRunner respectively. The right column of the Figure 1 demonstrates the performance drop $\mathcal{P}_w(p)$ with respect to action modification a_w for the RADIAL adversarially trained deep reinforcement learning policy proposed by Oikarinen et al. (2021) and the vanilla trained deep reinforcement learning policy in RoadRunner. Again the results in Figure 1 demonstrate that the vanilla training technique has better estimates for state-action values compared to the adversarial training method RADIAL, quite recently proposed by Oikarinen et al. (2021).

In particular, the curve for $\mathcal{P}_2(p)$ for RADIAL in RoadRunner lies well above the corresponding vanilla training curve. This implies that, while taking the second best action has a relatively mild effect on the vanilla-trained policy, it causes a dramatic loss in performance for RADIAL. Similarly, the $\mathcal{P}_w(p)$ curve for RADIAL in RoadRunner lies above the corresponding curve for the vanilla-trained

policy. This again implies that the vanilla-trained policy has a better estimate for which action will lead to lowest rewards than the RADIAL adversarially trained policy. The results reported in Figure 1 again demonstrate the loss of information in the state-action value function due to adversarial regulation of the temporal difference loss.

Figure 2 demonstrates that the overestimation bias discussed in the main body of our paper is again an issue for a newer adversarial training technique quite recently published in NeurIPS 2021. Furthermore, exactly as the previous adversarial training methods, RADIAL also learns inaccurate, inconsistent and overestimated state-action value functions. Hence, these results once more demonstrate the loss of information in the state-action value function as a novel fundamental trade-off intrinsic to adversarial training.

1.2 Supplementary Results on Inconsistencies in Action Ranking in Adversarially Trained Deep Neural Policies

As we mentioned in Section 6.1 of the main body of the paper the inaccuracies of the state-action value function reach a high enough level for the state-of-the-art adversarially trained deep neural policies such that the ranking of the sub-optimal actions is not correct anymore. This can be seen in Figure 3 in the $\mathcal{P}_2$ and $\mathcal{P}_w$ results. Note that $\mathcal{P}_2$ represents the performance drop (Definition 4.2) with action modification a_2, and $\mathcal{P}_w$ (Definition 4.2) represents the action modification with a_w.

Thus, it can be observed from Figure 3 that the performance drop $\mathcal{P}_2$ with action modification a_2 is higher than the performance drop $\mathcal{P}_w$ with action modification a_w. In more detail $\mathcal{P}_2$ 0.18257-dominates $\mathcal{P}_w$ in BankHeist (Definition 4.3). This demonstrates that the state-of-the-art adversarially trained deep neural policies are not ranking the sub-optimal actions correctly. Note that as we discussed in the main body of the paper in Section 6.1 this poses a problem for learning optimal state-action value functions Lin & Zhou (2020); Alshiekh et al. (2018).

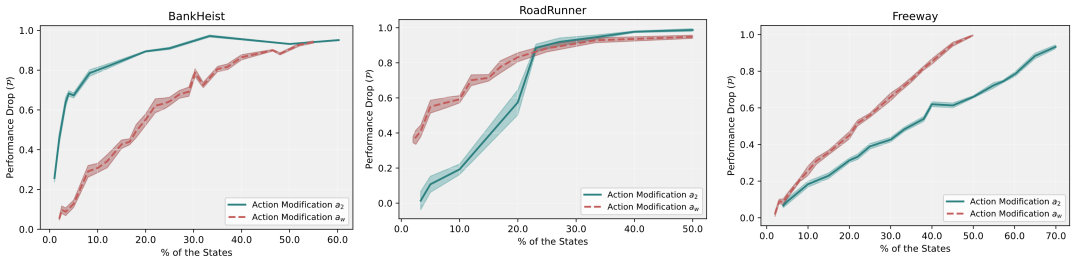


Figure 3: Consistency results for ranked actions via performance drop $\mathcal{P}_2$ and $\mathcal{P}_w$ for the state-of-the-art adversarially trained deep neural policies.

1.3 Overestimation of State-Action Values

In this section we provide supplementary results for the overestimation bias caused by state-of-the-art adversarially trained deep neural policies. In particular, in Section 6.3 of the main body of the paper we explained the problem of overestimation of state-action values. Furthermore, in Section 6.2 we empirically demonstrate that state-of-the-art adversarially trained deep neural policies overestimate the state-action values. In this section we further provide results on state-action values of the optimal action for vanilla and adversarially trained deep neural policies when p_{a_2} is equal to 0.1, 0.2 and 0.3 respectively. Note that in the main body of the paper we claim that the reason for this overestimation lies in the fact that the state-of-the-art deep neural policy adversarial training is solely an extension of adversarial training in image classification tasks, which is based on penalizing the wrong "label". However, this approach does not directly correspond to deep neural policies. The correct label in image classification can be connected to the optimal action in deep neural policies in this analogy. However, the wrong label does not correspond to sub-optimal actions. An optimal Q-function represents the discounted expected cumulative rewards received when taking an action a in state s. Hence, the sub-optimal actions have much more meaning in collecting rewards than solely misclassifying an image.

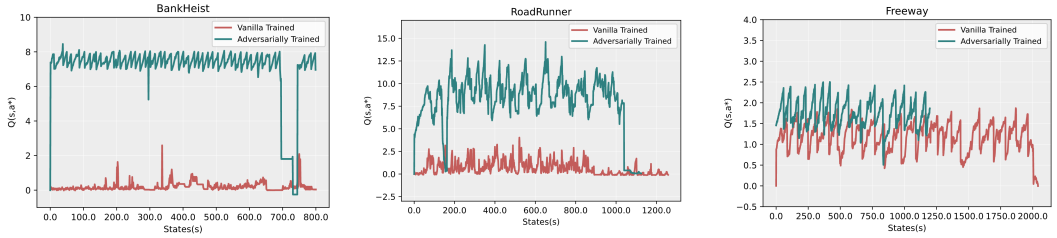


Figure 4: State-action values of the best action $Q(s, a^*)$ for vanilla trained deep neural policies and adversarially trained deep neural policies when p_{a_2} is 0.1.

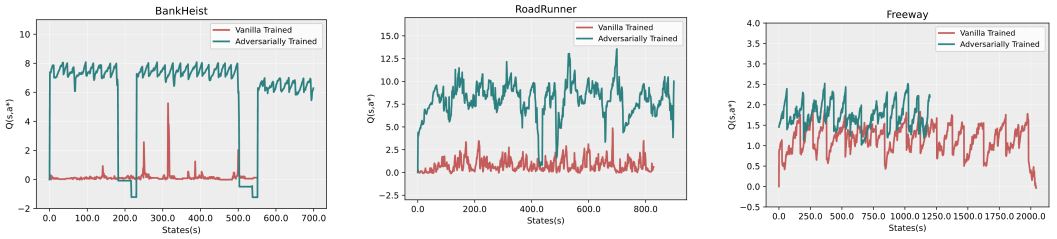


Figure 5: State-action values of the best action $Q(s, a^*)$ for vanilla trained deep neural policies and adversarially trained deep neural policies when p_{a_2} is 0.2.

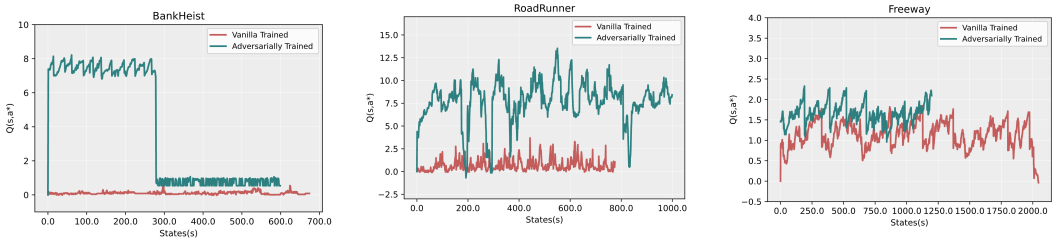


Figure 6: State-action values of the best action $Q(s, a^*)$ for vanilla trained deep neural policies and adversarially trained deep neural policies when p_{a_2} is 0.3.

1.4 Supplementary Results on Action Gap

In Section 6.4 of the main body of our paper we discuss the action gap phenomenon introduced by Farahmand (2011). Note that the action gap is defined as $\kappa(Q, s) = \max_{a' \in A} Q(s, a') - \max_{a \notin \arg\max_{a' \in A} Q(s,a')} Q(s, a)$. Further, we argue that both the existence of overestimation of state action values and the higher action gap in state-of-the-art adversarially trained deep neural policies demonstrates that the hypothesis of Bellemare et al. (2016) cannot be true.

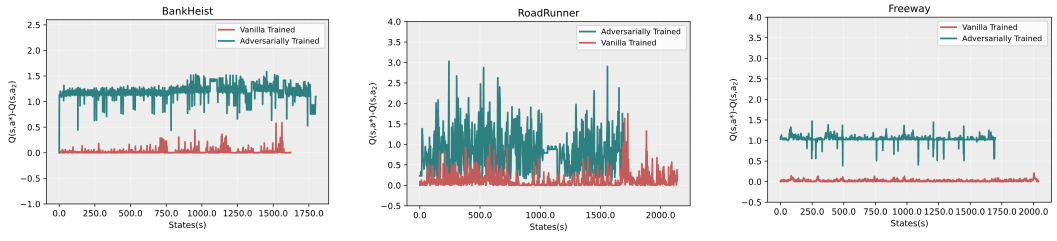


Figure 7: The action gap $Q(s, a^*) - Q(s, a_2)$ for the state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies for p_{a_2} is 0.

In this section we provide supplementary results on the action gap without the normalization $Q(s,a)/\sum_a |Q(s,a)|$. In particular, Figure 7, Figure 8 and Figure 9 show the action gap for the vanilla trained deep neural policies and state-of-the-art adversarial deep neural policies when p_{a_2} is 0, 0.1 and 0.2 respectively. Hence, the action gap for adversarially trained deep neural policies is higher than for vanilla trained deep neural policies.

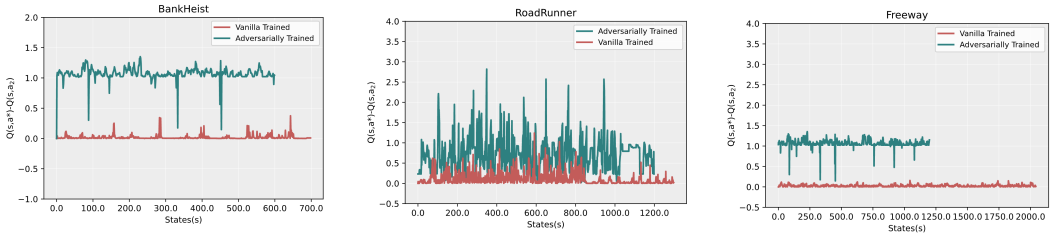


Figure 8: The action gap $Q(s,a^*) - Q(s,a_2)$ for state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies for p_{a_2} is 0.1.

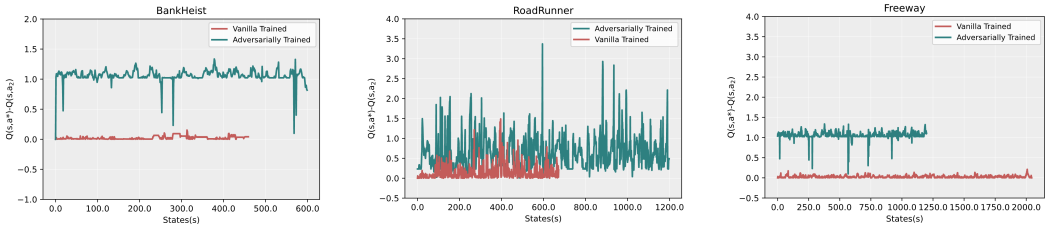


Figure 9: The action gap $Q(s,a^*) - Q(s,a_2)$ for state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies for p_{a_2} is 0.2.

1.5 Supplementary Results on Action Gap with Normalized State-Action Values

In the remainder of this section we provide additional results on normalized state-action values for adversarially trained and vanilla trained deep neural policies.

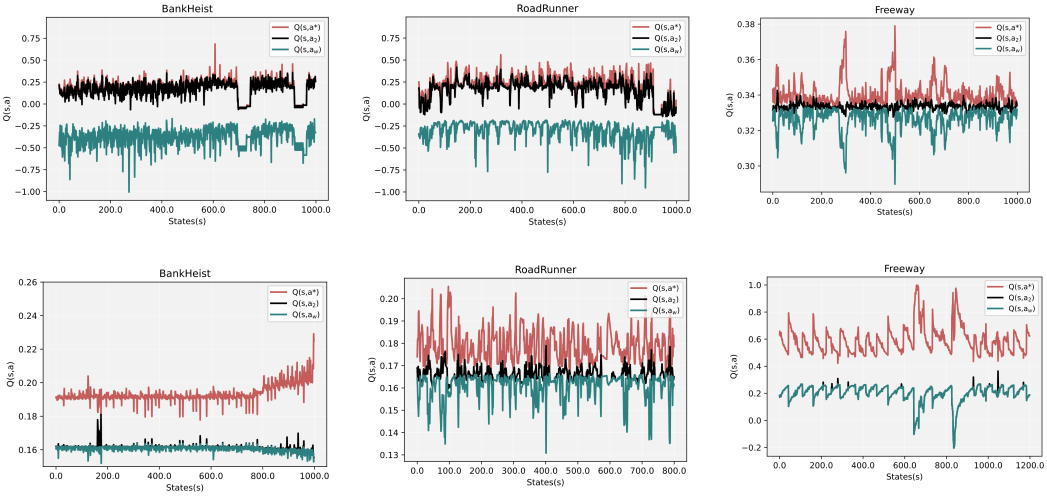


Figure 10: Normalized state-action values for the best action a^*, second best action a_2 and worst action a_w over states when p_{a_2} is 0.01. Row1: Vanilla trained deep neural policies. Row2: State-of-the-art adversarially trained deep neural policies.

In more detail, Figure 10 and Figure 11 show the normalized state-action values of the optimal action, second best action a_2 and worst action a_w for vanilla trained deep neural policies and adversarially trained deep neural policies when p_{a_2} is 0.01 and 0.1 respectively. Thus, Figure 10 and Figure 11 demonstrate that the action gap is higher for the state-of-the-art adversarially trained deep neural policies compared to vanilla trained deep neural policies. Note that the state-action values in Figure 10 and Figure 11 are normalized Q-values (i.e. normalized via $Q(s,a)/\sum_a |Q(s,a)|$).

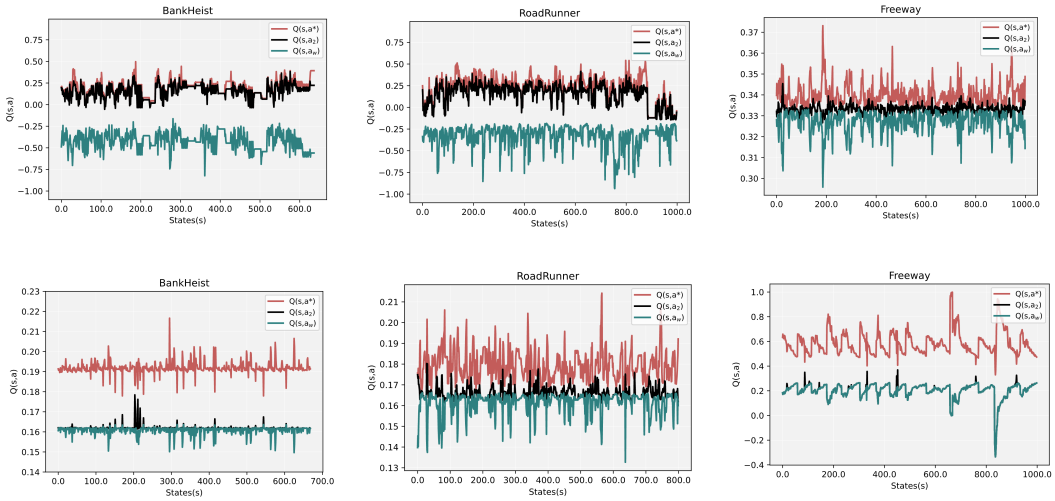


Figure 11: Normalized state-action values for the best action a^*, second best action a_2 and worst action a_w over states when p_{a_2} is 0.1. Row1: Vanilla trained deep neural policies. Row2: State-of-the-art adversarially trained deep neural policies.

1.6 Complete Proof of Theorem 3.4.

Theorem 1.1. *There is an MDP with linearly parameterized state-action values, optimal state-action value parameters θ^*, and a parameter vector θ such that: $\mathcal{L}(\theta) < \mathcal{L}(\theta^*)$, and the parameter vector θ overestimates the optimal state-action value and re-orders the sub-optimal ones.*

Proof. Let M be the MDP in the setting of Proposition 3.3 and define θ as in Proposition 3.3 by setting $\theta_1 = (1+\lambda)\theta_1^*$, $\theta_2 = (1+\lambda)\theta_2^*$, and $\theta_3 = (1-\lambda)\theta_3^*$. The overall regularized loss has the form

$$\mathcal{L}(\theta) = \mathcal{TD}(\theta) + \mathcal{R}(\theta).$$

Where $\mathcal{TD}(\theta)$ is the standard temporal difference loss. For the MDP M and parameters θ we can explicitly calculate this loss:

$$\begin{aligned}
\mathcal{TD}(\theta) &= \frac{1}{6}\sum_{i=1}^{2}\sum_{j=1}^{3}(r(s_i,a_j) + \gamma \max_k \langle \theta_k, s_{3-i}\rangle - \langle \theta_j, s_i\rangle)^2 \\
&\leq \frac{1}{6}\sum_{i=1}^{2}\sum_{j=1}^{3}(r(s_i,a_j) + \gamma \max_k (1+\lambda)\langle \theta_k^*, s_{3-i}\rangle - (1-\lambda)\langle \theta_j^*, s_i\rangle)^2 \\
&= \frac{1}{6}\sum_{i=1}^{2}\sum_{j=1}^{3}(r(s_i,a_j) + \gamma \max_k \langle \theta_k^*, s_{3-i}\rangle - \langle \theta_j^*, s_i\rangle + \lambda\gamma \max_k \langle \theta_k^*, s_{3-i}\rangle + \lambda\langle \theta_j^*, s_i\rangle)^2 \\
&= \frac{1}{6}\sum_{i=1}^{2}\sum_{j=1}^{3}(\lambda\gamma \max_k \langle \theta_k^*, s_{3-i}\rangle + \lambda\langle \theta_j^*, s_i\rangle)^2
\end{aligned}$$

where the final equality follows from the optimality of the paramters θ^*. Using the fact that $\langle \theta_j^*, s_i\rangle \leq 1$ for all i, j we conclude that

$$\mathcal{TD}(\theta) \leq (\gamma\lambda + \lambda)^2 < 4\lambda^2.$$

So for $\lambda < \frac{1}{4}$ we have by Proposition 3.3

$$\mathcal{TD}(\theta) \leq 4\lambda^2 < \lambda < \mathcal{R}(\theta^*) - \mathcal{R}(\theta).$$

Therefore $\mathcal{L}(\theta) < \mathcal{L}(\theta^*)$. Clearly, θ overestimates the optimal state-action values in both s_1 and s_2 by a factor of $1 + \lambda$. Furthermore, setting λ such that $\frac{1+\lambda}{1-\lambda} > \frac{\delta}{\eta}$ implies that a_3 will be the third ranked action in both states s_1 and s_2 i.e. that θ leads to re-ordering of the suboptimal actions. □

1.7 The fundamental trade-off between accurate estimation of Q-values and adversarial robustness

In this section we will prove that there is a fundamental trade-off between accurate estimation of Q-values and adversarial robustness. In particular, note that the goal of adversarial training is to ensure that a perturbation of magnitude ϵ to a state s will not result in a change to the action receiving the highest Q-value. Thus, a state-action value function $Q_\theta(s, a)$ is ϵ-robust if, for all s' such that $\|s - s'\|_2 < \epsilon$,

$$\text{argmax}_a Q(s, a) = \text{argmax}_a Q(s', a).$$

We will next construct an example in the setting of MDPs with linear function approximation where the optimal state-action value function Q^* is not robust, but there is a robust state-action value function Q_θ that overestimates the optimal state-action values.

Theorem 1.2. *Let $\epsilon > 0$. In the linear function approximation setting, there is an MDP such that all linear-state action value functions matching the optimal state-action values Q^* are not ϵ-robust. Furthermore, there is a linear state-action value function Q_θ that is ϵ-robust, but overestimates the optimal state-action values while maintaining the correct optimal action.*

Let there be two states s_1 and s_2 such that $\|s_1 - s_2\|_2 = 1$. Further suppose that the optimal state-action values satisfy $Q^*(s_1, a_1) = \epsilon/10$, $Q^*(s_1, a_2) = 0$, $Q^*(s_2, a_1) = 0.8$, and $Q^*(s_2, a_2) = 1.0$. Next let $Q_\theta(s, a)$ be any linearly parameterized state-action value function that agrees with $Q^*(s, a)$ on the states s_1 and s_2. Consider the one-dimensional functions $f_1(x) = Q_\theta((1-x) \cdot s_1 + x \cdot s_2, a_1)$ and $f_2(x) = Q_\theta((1-x) \cdot s_1 + x \cdot s_2, a_2)$ which are the restriction of $Q_\theta(s, a)$ to the line segment from s_1 to s_2. By linearity of Q_θ we also have that both f_1 and f_2 are linear. Furthermore, since Q_θ agrees with Q^* at s_1 and s_2, we know the values of both functions at two points i.e. $f_1(0) = Q^*(s_1, a_1)$, $f_1(1) = Q^*(s_2, a_1)$, $f_2(0) = Q^*(s_1, a_2)$, and $f_2(1) = Q^*(s_2, a_2)$. As f_1 and f_2 are linear functions on $\mathbb{R}$, the values at two points are sufficient to uniquely determine the functions. In particular we have

$$f_1(x) = (0.8 - \epsilon/10)x + \epsilon/10$$

$$f_2(x) = x$$

Note that these two lines intersect at the point $\hat{x} = \frac{\epsilon}{2+\epsilon}$. Let $\hat{s} = (1-\hat{x}) \cdot s_1 + \hat{x} \cdot s_2$. Since the lines of f_1 and f_2 intersect at $\hat{x}$, we conclude that $Q_\theta(\hat{s}, a_2) \geq Q_\theta(\hat{s}, a_1)$. However, $Q_\theta(s_1, a_1) > Q_\theta(s_1, a_2)$. Furthermore, $\|s_1 - \hat{s}\| = \frac{\epsilon}{2+\epsilon} < \epsilon$. Thus, Q_θ is not ϵ-robust.

However, if we instead choose new parameters θ' for the state-action value function so that $Q_{\theta'}(s_1, a_1) = 0.8$ and $Q_{\theta'}(s_1, a_2) = 0.7$ one can easily check that $Q_{\theta'}$ is ϵ-robust for all $\epsilon < 0.1$. Furthermore, observe that $Q_{\theta'}$ gives the correct ranking of actions in state s_1, but overestimates the optimal state-action value by a factor of $8/\epsilon$.

1.8 Further Experiments on the Linearly Parametrized MDP

To complement the theoretical results, we numerically optimized both the regularized and un-regularized loss function for the example MDP with linearly parameterized state-action values constructed in Section 3. Figure 12 demonstrates the state-action value function for each of the states the best action a^*, second best action a_2 and worst action a_w for the actions a_1, a_2, a_3. Note that the numerical optimization of the un-regularized (i.e. vanilla training) loss converges to the true optimal state-action values computed analytically in Section 3. Thus, the results reported in Figure 12 further demonstrate that the addition of the certified training regularizer leads to overestimation of the optimal state-action value function, and re-ordering of the suboptimal actions.

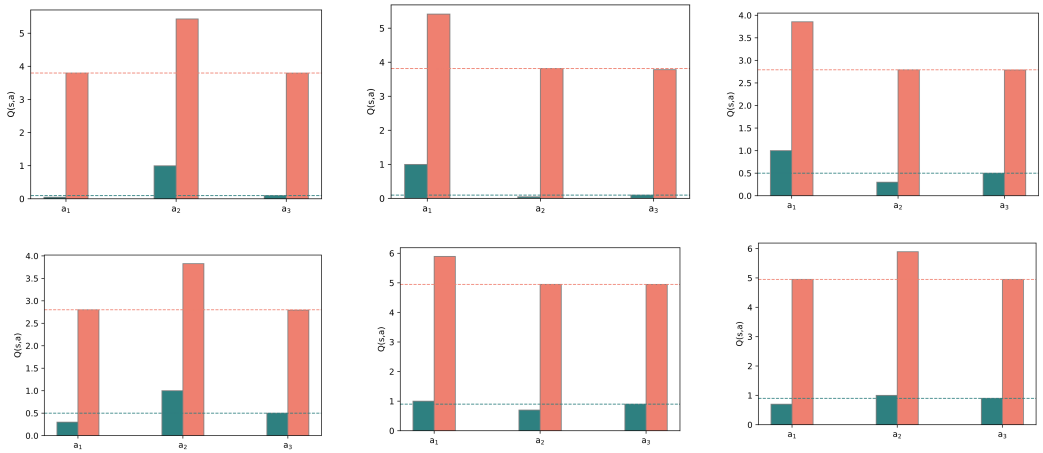


Figure 12: State-action values for the best action a^*, second best action a_2 and worst action a_w for the adversarially trained and vanilla trained deep neural policy loss function for the example MDP with linearly parameterized state-action values constructed in Section 3.

1.9 What does it entail to learn inaccurate, overestimated and inconsonant state-action values?

The fact that our paper explicitly theoretically and empirically demonstrates that certified adversarially trained policies learn inconsonant and inaccurate state-action values further implies significant concerns on the alignment with human decisions. The claim made in our paper regarding alignment with human decisions implies that human decision-making can allocate correct values for the sub-optimal actions. For concrete evidence on the human decision making process and the fact that humans have a better than random perception of actions that they do not take, please see (Wunderlich et al., 2009; Phillips et al., 2019; Hoeck et al., 2015).

Also further note that, as also initially described in the main body of our paper in Section 2.3, recent work demonstrated vulnerabilities of certified robust reinforcement learning policies from black-box adversarial attacks (Korkmaz, 2022) to natural attacks that revealed the generalization problems of adversarially trained deep reinforcement learning policies when compared to straightforward reinforcement learning (Korkmaz). While these studies highlight the safety and security problems in certified adversarially trained policies, our paper dives into and explains the particular reasons why adversarial training experiences these safety problems. We believe it is crucial to understand the root causes of these problems regarding AI-safety, because releasing models with guaranteed safety certifications with undiscovered non-robustness and vulnerabilities will in fact have serious consequences in the real world (Post, 2023; Guardian, 2022; Times, 2023).

1.10 Implementation Details

Note that to be able to provide a fair comparison State-Adversarial Double Deep Q-Network and Double Deep Q-Network are the exact same implementations described in the SA-DDQN paper described in Section 3 and Wang et al. (2016) respectively. In more detail for Double Deep Q-Network the batch size is 32, discount factor γ is 0.99, buffer size 50000, learning rate is 5×10^{-5} for the Adam optimizer, and random action probability is 0.02. Note that experience replay Schaul et al. (2016) is utilized. More details can be found in Dhariwal et al. (2017) and Wang et al. (2016) on Double Deep Q-Networks. The state-of-the-art adversarial deep neural policy is the exact same implementation as in the SA-DDQN paper. Adversarial deep neural policies are trained via experience replay as well Schaul et al. (2016). Note that State-Adversarial Double Deep Q-Network is trained via the regularizer $\mathcal{R}(\theta) = \sum_s \left(\max_{\bar{s} \in D_\epsilon(s)} \max_{a \neq a^*(s)} Q_\theta(\bar{s}, a) - Q_\theta(\bar{s}, a^*(s)) \right)$ where $a^*(s) = \arg\max_a Q(s, a)$ inside ϵ-ball $D_\epsilon(s) = \{\bar{s} : \|s - \bar{s}\|_\infty \leq \epsilon\}$. Hence, this ϵ is set to $1/255$. Note that the regularization is added to the temporal difference loss in the Q-update. The regularization parameter of state-adversarial is $\kappa \in \{0.005, 0.01, 0.02\}$. The initial 1.5×10^6 frames are trained without regularization.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 2669–2678. AAAI Press, 2018.

Marc G. Bellemare, Georg Ostrovski, Arthur Guez, Philip S. Thomas, and Rémi Munos. Increasing the action gap: New operators for reinforcement learning. In Dale Schuurmans and Michael P. Wellman (eds.), *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 1476–1483. AAAI Press, 2016.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

Amir Massoud Farahmand. Action-gap phenomenon in reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

The Guardian. Tesla behind eight-vehicle crash was in 'full self-driving' mode, says driver. December 2022.

Nicole Van Hoeck, Patrick D. Watson, and Aron K. Barbey. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in Human Neuroscience*, 2015.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*.

Ezgi Korkmaz. Deep reinforcement learning policies learn shared adversarial features across mdps. *AAAI Conference on Artificial Intelligence*, 2022.

Kaixiang Lin and Jiayu Zhou. Ranking policy gradient. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Tuomas P. Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. Robust deep reinforcement learning through adversarial loss. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 26156–26167, 2021.

Jonathan Phillips, Adam Morris, and Fiery Cushman. How we know what not to think. *Trends in Cognitive Sciences*, 2019.

The Washington Post. Cruise recalls all its driverless cars after pedestrian hit and dragged. November 2023.

Tom Schaul, John Quan, Ioannis Antonogloua, and David Silver. Prioritized experience replay. *International Conference on Learning Representations (ICLR)*, 2016.

New York Times. Driverless taxis blocked ambulance in fatal accident, san francisco fire department says. September 2023.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando. De Freitas. Dueling network architectures for deep reinforcement learning. *Internation Conference on Machine Learning ICML.*, pp. 1995–2003, 2016.

Klaus Wunderlich, Antonio Rangel, and John P. O'Doherty. Neural computations underlying action-based decision making in the human brain. *Proceedings of the National Academy of Sciences (PNAS)*, 2009.