JUXTALIGN: A FOUNDATIONAL ANALYSIS ON ALIGNMENT OF CERTIFIED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential decision making in highly complex MDPs with high-dimensional observations and state dynamics became possible with the progress achieved in deep reinforcement learning research. At the same time, deep neural policies have been observed to be highly unstable with respect to the minor sensitivities in their state space induced by non-robust directions. To alleviate these volatilities a line of work suggested techniques to cope with this problem via explicitly regularizing the temporal difference loss for the worst-case sensitivity. In this paper we provide theoretical foundations on the failure instances of the approaches proposed to overcome instabilities of the deep neural policy manifolds. Our comprehensive analysis reveals that certified reinforcement learning learns misaligned values. Our empirical analysis in the Arcade Learning Environment further demonstrates that the state-of-the-art certified policies learn inconsistent and overestimated value functions compared to standard training techniques. In connection to this analysis, we highlight the intrinsic gap between how natural intelligence understands and interacts with an environment in contrast to policies learnt via certified training. This intrinsic gap between natural intelligence and the restrictions induced by certified training on the capabilities of artificial intelligence further demonstrates the need to rethink the approach in establishing reliable and aligned deep reinforcement learning policies.

1 INTRODUCTION

Inspired by the learning dynamics and cognitive abilities of natural intelligence (Watkins, 1989; Kehoe et al., 1987; Romo & Schultz, 1990; Montague et al., 1996; Schultz et al., 1993; Pan et al., 2005), reinforcement learning research has been the focal point of immense research progress (Mnih et al., 2015; Hasselt et al., 2016b). Deep reinforcement learning has become an emerging field in the past decade with the introduction of deep neural networks as function approximators leading to learning policies that can surpass human cognitive abilities in highly complicated tasks by solely interacting with a given environment through trial and error (Mnih et al., 2015; Kapturowski et al., 2023).

Along with the strong inspiration from neuroscience, remarkably reinforcement learning further comes with mathematically provable guarantees on what can be learnt asymptotically (Sutton, 1984; Watkins & Dayan, 1992). A recent line of research highlighted the safety concerns of reinforcement learning, and further proposed a line of algorithms that modify standard reinforcement learning algorithms to ensure reliability and robustness in deep reinforcement learning (Madry et al., 2018; Korkmaz, 2024).

At the same time, recent research in neuroscience has been able to identify structures in the human brain that directly compute counterfactual action-values, and then compare these values in order to make decisions. In particular, recent work in decision neuroscience demonstrated that while the prefrontal cortex of natural intelligence records the expected value of the actions executed, the dorsomedial frontal cortex analyzes counterfactual decisions of the human brain (Wunderlich et al., 2009; Lau & Glimcher, 2007; Klein-Flügge et al., 2016).

In this paper, we analyze the effects of safety in reinforcement learning and our analysis discovers that the line of research focused on safety fails to deliver the guarantees implied by *"certified safety and robustness"*, and further risks potentially significant changes to the behavior and semantics of the trained policies, particularly in how they align with how natural intelligence reasons about the values of actions.

Essentially in this paper we aim to seek answers for the following questions: *(i) What is the intrinsic alignment between natural intelligence decision making and reinforcement learning?*, *(ii) Do our efforts on ensuring safety divert the original neuroscientific motivations of reinforcement learning algorithms?* To be able to answer these questions we focus on the foundations of reinforcement learning and its alignment with natural intelligence, and make the following contributions:

- We introduce a theoretically well-founded analysis of the state-action value function learnt by state-of-the-art certified adversarial training and standard reinforcement learning. Our paper is the first one that demonstrates, both theoretically and empirically, that certified robust training has manifold flaws, and security and safety issues that do not match its promises.
- We highlight the connection between neural correlates of action values in natural intelligence and understanding deep neural policy decision making. In particular, our analysis reveals that robust training methods learn policies that are misaligned with human decision making processes, in which humans have a better than random perception of actions that they do not take. Furthermore, our results demonstrate that standard reinforcement learning in fact captures the values of counterfactual actions while robust training methods cannot.
- We conduct experiments in MDPs with high-dimensional state spaces from the Arcade Learning Environment (ALE). Our comprehensive systematic analysis demonstrates that vanilla deep neural policies learn values for decisions that are highly close to how natural intelligence assigns values for actions, yet further orthogonal to how certified training makes decisions. Thence these results demonstrate that standard reinforcement learning learns a more accurate and stable representation of the state-action value function compared to the state-of-the-art adversarially trained deep neural policies.
- Our paper further provides foundations and demonstrates that there is an intrinsic trade-off between accurate estimation of state-action values and robustness. Our comprehensive and systematic analysis reveals the loss of information in the state-action value function as a novel fundamental trade-off intrinsic to certified training.

2 Background and Preliminaries

Neuroscientific Results and Alignment with Natural Intelligence Decisions Making: The fact that natural intelligence assigns meaningful values to counterfactual actions is a well-studied phenomenon in neuroscience (Wunderlich et al., 2009; Lee et al., 2012; Phillips et al., 2019). In particular, human cognitive decision making assigns counterfactual values to decisions not taken, and uses these values to inform future decision making. Furthermore, humans do preserve the knowledge on the correct ordering of both factual and counterfactual decisions (Hoeck et al., 2015; Phillips et al., 2019; Grabenhorst & Rolls, 2011). Notably, the results in Figure 1 report analysis of fMRI scans of human brains during a decision-making task to identify a neural structure that compares the values of chosen and unchosen options for a particular decision. The results demonstrate that the value of each option was encoded in this structure, and that the actual decisions made were correlated with these values (Klein-Flügge et al., 2016).

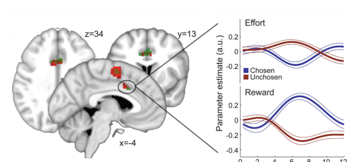

Figure 1: Human decision making and value assignment for options (Klein-Flügge et al., 2016).

Our extensive analysis and results discover that current robust training methods move artificial intelligences further out of alignment with natural intelligence by systematically disrupting the information on the values of counterfactual actions to be nearly random. We believe that such misalignment provides evidence that certified training methods are insufficient to resolve the robustness and safety problems of current artificial intelligence, and further portrays the dichotomy between certified training and natural intelligence.

Preliminaries In deep reinforcement learning the goal is to learn a policy for taking actions in a Markov Decision Process (MDP) that maximize discounted expected cumulative reward. An MDP is represented by a tuple $\mathcal{M} = (S, \mathcal{A}, P, r, \rho_0, \gamma)$ where S is a set of continuous states, $\mathcal{A}$ is a discrete set of actions, P is a transition probability distribution on $S \times \mathcal{A} \times S$, $r : S \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function, ρ_0 is the initial state distribution, and γ is the discount factor. The objective in reinforcement learning is to learn a policy $\pi : S \rightarrow P(\mathcal{A})$ which maps states to probability distributions on actions in order to maximize the expected cumulative reward $R = \mathbb{E} \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$ where $a_t \sim \pi(s_t)$. In $\mathcal{Q}$-learning (Watkins, 1989) the goal is to learn the optimal state-action value function $\mathcal{Q}^*(s, a) = R(s, a) + \sum_{s' \in S} P(s'|s, a) \max_{\hat{a} \in \mathcal{A}} \mathcal{Q}^*(s', \hat{a})$. Thus, the optimal policy is determined by choosing the action $a^*(s) = \arg\max_a \mathcal{Q}(s, a)$ in state s.

Adversarial Crafting and Training: Szegedy et al. (2014) observed that imperceptible perturbations could change the decision of a deep neural network and proposed a box constrained optimization method to produce such perturbations. Goodfellow et al. (2015) suggested a faster method to produce such perturbations based on the linearization of the cost function used in training the network. Kurakin et al. (2016) proposed the iterative version of the fast gradient sign method proposed by Goodfellow et al. (2015) inside an ϵ-ball

$$x_{\text{adv}}^{N+1} = \text{clip}_\epsilon(x_{\text{adv}}^N + \alpha \text{sign}(\nabla_x J(x_{\text{adv}}^N, y))) \tag{1}$$

in which $J(x, y)$ represents the cost function used to train the deep neural network, x represents the input, and y represents the output labels. While several other methods have been proposed (e.g. Korkmaz (2020)) using a momentum-based extension of the iterative fast gradient sign method,

$$v_{t+1} = \mu \cdot v_t + \frac{\nabla_{s_{\text{adv}}} J(s_{\text{adv}}^t + \mu \cdot v_t, a)}{\|\nabla_{s_{\text{adv}}} J(s_{\text{adv}}^t + \mu \cdot v_t, a)\|_1} \;,\; s_{\text{adv}}^{t+1} = s_{\text{adv}}^t + \alpha \cdot \frac{v_{t+1}}{\|v_{t+1}\|_2}$$

adversarial training has mostly been conducted with perturbations computed by projected gradient descent (PGD) proposed by Madry et al. (2018) (i.e. Equation 1).

Adversaries, Robustness and Certified Training in Deep Neural Policies: The initial investigation on resilience of deep neural policies was conducted by Kos & Song (2017) and Huang et al. (2017) concurrently based on the utilization of the fast gradient sign method proposed by Goodfellow et al. (2015). Recent work demonstrated that deep reinforcement learning policies learn shared adversarial features across MDPs revealing an underlying linear structure learnt by the deep reinforcement learning policies (Korkmaz, 2022; 2024). While several studies focused on improving optimization techniques to compute optimal perturbations, a line of research focused on making deep neural policies resilient to these perturbations. In particular, Pinto et al. (2017) proposed to model the dynamics between the adversary and the deep neural policy as a zero-sum game where the goal of the adversary is to minimize expected cumulative rewards of the deep neural policy. Gleave et al. (2020) approached this problem with an adversary model which is restricted to take natural actions in the MDP instead of modifying the observations with ℓ_p-norm bounded perturbations. The authors model this dynamic as a zero-sum Markov game and solve it via self play. Recently, Huan et al. (2020) proposed to model this interaction between the adversary and the deep neural policy as a state-adversarial MDP, and claimed that their proposed algorithm State Adversarial Double Deep Q-Network (SA-DDQN) learns theoretically certified robust policies against natural noise and perturbations. Recent work demonstrated that certified training learns identical high-sensitivity directions with standard training, thence can be attacked with a black-box approach (Korkmaz, 2022). Furthermore, some studies showed that certified training while not able to generalize compared to vanilla training, furthermore learns non-robust directions that are more unstable with larger oscillations (Korkmaz, 2024). Yet none of these studies provided foundational explanations on why such a promising and theoretically well-founded line of algorithms were in fact doomed to fail.

3 The Orthogonality of Natural Intelligence Decision Making and Adversarial Training

The theoretically motivated adversarial, i.e. certified robust, training techniques achieve certified defense against adversarial perturbations inside the ϵ-ball $\mathcal{D}_\epsilon(s) = \{\bar{s} : \|s - \bar{s}\|_\infty \leq \epsilon\}$. However,

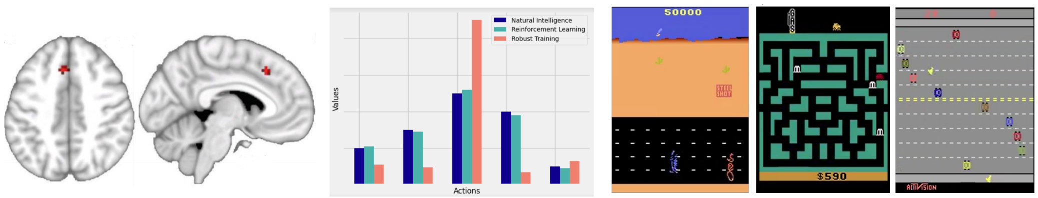

Figure 2: Representation of the misalignment between natural intelligence and robust training, and the alignment between reinforcement learning and natural intelligence.

we provide foundational evidence that this approach induces significant changes in the $\mathcal{Q}$-function where the state-action value function no longer accurately represents the MDP. In particular, robust training causes deep neural policies to learn overestimated state-action values, and furthermore the $\mathcal{Q}$-values for non-optimal actions are reduced in accuracy to the point where their relative ranking changes. Furthermore, we connect and highlight the neural processing of decision making of natural intelligence and certified training (Wunderlich et al., 2009; Lau & Glimcher, 2007; Grabenhorst & Rolls, 2011). Our results demonstrate that certified training constructs policies that are disjoint and orthogonal to natural intelligence decision making. The fundamental theoretical basis for adversarial training techniques comes from Danskin's theorem.

Theorem 3.1 (Danskin (1967))**.** *Let $\mathcal{X}$ be a compact topological space $f : \mathbb{R}^n \times X \to \mathbb{R}$, $f(\cdot, x)$ is differentiable for every $x \in \mathcal{X}$, $x^*(\theta) = \{x \in \arg\max_{x \in \mathcal{X}} f(\theta, x)\}$ and $\nabla_\theta f(\theta, x)$ is continuous on $\mathbb{R}^n \times \mathcal{X}$. Then the max function $\kappa(\theta) = \max_{x \in \mathcal{X}} f(\theta, x)$ is locally Lipschitz continuous, directionally differentiable, and its directional derivatives satisfy $\kappa'(\theta, h) = \sup_{x \in x^*(\theta)} h^\top \nabla_\theta f(x, \theta)$. Furthermore, if the set $x^*(\theta)$ has size one i.e. there is a unique maximizer x^*_θ then $\nabla_\theta \kappa(\theta) = \nabla_\theta f(\theta, x^*_\theta)$.*

In particular, Danskin's theorem gives a method to compute the gradient of a function that is defined in terms of a maximization over some set. With this theoretically well-motivated start a line of algorithms have been proposed to make models robust including deep reinforcement learning (i.e. Section 2). The basic approach of certified (i.e. adversarial) training techniques is based on adding a regularizer to the standard $\mathcal{Q}$-learning update. The regularizer is designed to penalize $\mathcal{Q}$-functions for which a perturbed state $\bar{s} \in \mathcal{D}_\epsilon(s)$ can change the identity of the highest $\mathcal{Q}$-value action. For the baseline adversarial training technique (Huan et al., 2020) we will theoretically analyze the effects of this regularizer.

Definition 3.2 (*Baseline Adversarial Training*)**.** The regularizer to achieve certified robustness for $\mathcal{Q}_\theta(s, a)$ $\forall \bar{s} \in \mathcal{D}_\epsilon(s)$ is given by

$$\mathcal{R}(\theta) = \sum_s \left(\max_{\bar{s} \in \mathcal{D}_\epsilon(s)} \max_{a \neq \arg\max_a \mathcal{Q}_\theta(s,a)} \mathcal{Q}_\theta(\bar{s}, a) - \mathcal{Q}_\theta(\bar{s}, \arg\max_a \mathcal{Q}_\theta(s, a)) \right).$$

The adversarial training algorithm proceeds by adding $\mathcal{R}(\theta)$ to the standard temporal difference loss used in DQN $\mathcal{L}(\theta) = \mathcal{L}_\mathcal{H}\left(r(s, a) + \gamma \max_{a'} \mathcal{Q}^{\text{target}}(s', a') - \mathcal{Q}_\theta(s, a)\right) + \mathcal{R}(\theta)$.

In the remainder of this section we will provide the theoretical foundations on: *(i) how certified training produces policies that are completely orthogonal to natural intelligence decision making*, and *(ii) why this promising line of algorithms has failed to deliver its promises*. Let us now describe the construction of an MDP $\mathcal{M}$ where the use of the regularizer causes randomized decision making $\forall a \in \mathcal{A}_s^\perp$ where $\mathcal{A}_s^\perp := \{a | a \neq \arg\max_{\hat{a}} \mathcal{Q}(s, \hat{a})\}$, and overestimation of the state-action values $\forall a \in \mathcal{A}$. There are two states parametrized by feature vectors $s_1, s_2 \in \mathbb{R}^n$, and there are three possible actions $\{a_i\}_{i=1}^3$ in each state. Taking any of the three actions in state s_1 leads to a transition to state s_2 and vice versa. Let $1 > \gamma > 0$ be the discount factor, and let $\delta > \eta > 0$ be small constants with $\gamma > \delta$. The rewards for each action are as follows: $r(s_1, a_1) = 1 - \gamma$, $r(s_1, a_2) = \eta - \gamma$, $r(s_1, a_3) = \delta - \gamma$, $r(s_2, a_1) = \eta - \gamma$, $r(s_2, a_2) = 1 - \gamma$, and $r(s_2, a_3) = \delta - \gamma$. Clearly, the optimal policy is to always take action a_1 in state s_1, and action a_2 in state s_2 as these are the only actions giving positive reward. Thus the optimal state-action values are given by: $\mathcal{Q}^*(s_1, a_1) = \mathcal{Q}^*(s_2, a_2) = \sum_{t=0}^\infty (1 - \gamma)\gamma^t = 1$, $\mathcal{Q}^*(s_1, a_2) = \mathcal{Q}^*(s_2, a_1) = \eta - \gamma + \gamma \sum_{t=0}^\infty (1 - \gamma)\gamma^t = \eta$, and $\mathcal{Q}^*(s_1, a_3) = \mathcal{Q}^*(s_2, a_3) = \delta - \gamma + \gamma \sum_{t=0}^\infty (1 - \gamma)\gamma^t = \delta$. Let the $\mathcal{Q}$-function be linearly parametrized by $\theta = (\theta_1, \theta_2, \theta_3)$ so that $\mathcal{Q}_\theta(s, a_i) = \langle \theta_i, s \rangle$. Finally, let Φ_i for $i \in \{1, 2, 3\}$ be three orthonormal vectors, and let the state feature vectors satisfy:

$$1.\ s_1 = \Phi_1 + \delta\Phi_3 + \eta\Phi_2 \quad \text{and} \quad 2.\ s_2 = \Phi_2 + \delta\Phi_3 + \eta\Phi_1$$

Then it follows that the optimal $\mathcal{Q}$-function is parametrized by $\theta^* = (\theta_1^*, \theta_2^*, \theta_3^*)$ where $\theta_i^* = \Phi_i$ i.e. $\mathcal{Q}_{\theta^*}(s,a) = \mathcal{Q}^*(s,a)$ for all s and a. Thus, according to the function $\mathcal{Q}_{\theta^*}(s,a)$, for s_1 the best action is a_1, for s_2 the best action is a_2, and in all states the second-best action is a_3. Next we identify the optimal perturbations used in the computation of the regularizer $\mathcal{R}(\theta^*)$ for this setting.

Proposition 3.3. *In the MDP $\mathcal{M}$ for any $\epsilon > 0$.*

1. *For* $s = s_1 : s + \dfrac{\epsilon}{\sqrt{2}}(\theta_3^* - \theta_1^*) = \underset{\bar{s} \in \mathcal{D}_\epsilon(s)}{\arg\max} \max_{a \neq a^*(s)} \mathcal{Q}_{\theta^*}(\bar{s}, a) - \mathcal{Q}_{\theta^*}(\bar{s}, a^*(s))$

2. *For* $s = s_2 : s + \dfrac{\epsilon}{\sqrt{2}}(\theta_3^* - \theta_2^*) = \underset{\bar{s} \in \mathcal{D}_\epsilon(s)}{\arg\max} \max_{a \neq a^*(s)} \mathcal{Q}_{\theta^*}(\bar{s}, a) - \mathcal{Q}_{\theta^*}(\bar{s}, a^*(s))$

Proof. We will prove item 1, and item 2 will follow from an identical argument with roles of θ_1^* and θ_2^* swapped. Let $s = s_1$. Since $a^*(s) = 1$, there are two case to consider for the maximum over $a \neq a^*(s)$, either $a = 2$ or $a = 3$. In the case that $a = 2$ we have

$$\max_{\bar{s} \in \mathcal{D}_\epsilon(s)} \mathcal{Q}_{\theta^*}(\bar{s}, a) - \mathcal{Q}_{\theta^*}(\bar{s}, a^*(s)) = \max_{\bar{s} \in \mathcal{D}_\epsilon(s)} \langle \theta_2^*, \bar{s} \rangle - \langle \theta_1^*, \bar{s} \rangle. \tag{2}$$

This is the maximum in a ball of radius ϵ around s of the linear function $\langle \theta_2^* - \theta_1^*, \bar{s} \rangle$. Therefore the maximum is achieved by $\bar{s} = s + \frac{\epsilon}{\sqrt{2}}(\theta_2^* - \theta_1^*)$. The corresponding maximum value is

$$\max_{\bar{s} \in \mathcal{D}_\epsilon(s)} \langle \theta_2^*, \bar{s} \rangle - \langle \theta_1^*, \bar{s} \rangle = \langle \theta_2^* - \theta_1^*, s \rangle + \epsilon \|\theta_2^* - \theta_1^*\|_2 = \eta - 1 + \epsilon\sqrt{2}. \tag{3}$$

In the case that $a = 3$ an identical argument implies that the maximum is achieved by $\bar{s} = s + \frac{\epsilon}{\sqrt{2}}(\theta_3^* - \theta_1^*)$, with corresponding maximum value

$$\max_{\bar{s} \in \mathcal{D}_\epsilon(s)} \langle \theta_3^*, \bar{s} \rangle - \langle \theta_1^*, \bar{s} \rangle = \langle \theta_3^* - \theta_1^*, s \rangle + \epsilon \|\theta_3^* - \theta_1^*\|_2 = \delta - 1 + \epsilon\sqrt{2}. \tag{4}$$

Because $\delta > \eta$ we conclude that the value achieved in 4 is larger than that in 3. Thus the maximizer is $\bar{s} = s + \frac{\epsilon}{\sqrt{2}}(\theta_3^* - \theta_1^*)$ as desired. $\square$

In words, the optimal direction to perturb the state s_1 in order to have $a^*(s) \neq a^*(\bar{s})$ is toward $\theta_3^* - \theta_1^*$. Similarly for the state s_2, the optimal perturbation is toward $\theta_3^* - \theta_2^*$. Next we use this fact to show that in order to decrease the regularizer it is sufficient to simply increase the magnitude of θ_1 and θ_2, and decrease the magnitude of θ_3.

Proposition 3.4. *In the MDP $\mathcal{M}$ let $\lambda > 0$ and suppose that $(1-\lambda)\delta < (1+\lambda)\eta < \delta$. Let $\theta = (\theta_1, \theta_2, \theta_3)$ be given by $\theta_1 = (1+\lambda)\theta_1^*$, $\theta_2 = (1+\lambda)\theta_2^*$ and $\theta_3 = (1-\lambda)\theta_3^*$. Then $\mathcal{R}(\theta) < \mathcal{R}(\theta^*)$.*

The proof is provided in the supplementary material. Combining Proposition 3.4 and Proposition 3.3 we can prove the main result of this section on the effects of worst-case regularization on the state-action value function.

Theorem 3.5 (*Existence of Overestimation and Misalignment of Counterfactual Decisions*)**.** *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 $\mathcal{M}$ 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 $\mathcal{TD}(\theta)\leq(\gamma\lambda+\lambda)^2<4\lambda^2$. Thus, 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. $\square$

Next we will prove that there is a fundamental trade-off between accurate estimation of $\mathcal{Q}$-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 $\mathcal{Q}$-value. Thus, formally the canonical definition of ϵ-robustness in deep reinforcement learning is

Definition 3.6 (*ϵ-robust deep neural policy*)**.** A state-action value function $\mathcal{Q}_\theta(s,a)$ is ϵ-robust if $\operatorname{argmax}_a\mathcal{Q}(s,a)=\operatorname{argmax}_a\mathcal{Q}(\bar{s},a)$, for all $\bar{s}\in\mathcal{D}_\epsilon(s)$ such that $\|s-\bar{s}\|_2<\epsilon$.

We will next demonstrate the instances of MDPs with linear function approximation where the optimal state-action value function $\mathcal{Q}^*$ is not robust, but there is a robust state-action value function $\mathcal{Q}_\theta$ that overestimates the optimal state-action values.

Theorem 3.7 (*Intrinsic trade-off between overestimation and robustness*)**.** *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 $\mathcal{Q}^*$ are not ϵ-robust. Furthermore, there is a linear state-action value function $\mathcal{Q}_\theta$ that is ϵ-robust, but overestimates the optimal state-action values while maintaining the correct optimal action.*

Proof. 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 $\mathcal{Q}^*(s_1,a_1)=\epsilon/10$, $\mathcal{Q}^*(s_1,a_2)=0$, $\mathcal{Q}^*(s_2,a_1)=0.8$, and $\mathcal{Q}^*(s_2,a_2)=1.0$. Next let $\mathcal{Q}_\theta(s,a)$ be any linearly parameterized state-action value function that agrees with $\mathcal{Q}^*(s,a)$ on the states s_1 and s_2. Consider the one-dimensional functions $\Psi_1(\xi)=\mathcal{Q}_\theta((1-\xi)\cdot s_1+\xi\cdot s_2,a_1)$ and $\Psi_2(\xi)=\mathcal{Q}_\theta((1-\xi)\cdot s_1+\xi\cdot s_2,a_2)$ which are the restriction of $\mathcal{Q}_\theta(s,a)$ to the line segment from s_1 to s_2. By linearity of $\mathcal{Q}_\theta$ we also have that both Ψ_1 and Ψ_2 are linear. Furthermore, since $\mathcal{Q}_\theta$ agrees with $\mathcal{Q}^*$ at s_1 and s_2, we know the values of both functions at two points i.e. $\Psi_1(0)=\mathcal{Q}^*(s_1,a_1)$, $\Psi_1(1)=\mathcal{Q}^*(s_2,a_1)$, $\Psi_2(0)=\mathcal{Q}^*(s_1,a_2)$, and $\Psi_2(1)=\mathcal{Q}^*(s_2,a_2)$. As Ψ_1 and Ψ_2 are linear functions on $\mathbb{R}$, the values at two points are sufficient to uniquely determine the functions. In particular we have

$$\Psi_1(\xi)=(0.8-\epsilon/10)\xi+\epsilon/10 \quad\text{and}\quad \Psi_2(\xi)=\xi$$

Note that these two lines intersect at the point $\hat{\xi}=\frac{\epsilon}{2+\epsilon}$. Let $\hat{s}=(1-\hat{\xi})\cdot s_1+\hat{\xi}\cdot s_2$. Since the lines of Ψ_1 and Ψ_2 intersect at $\hat{\xi}$, we conclude that $\mathcal{Q}_\theta(\hat{s},a_2)\geq\mathcal{Q}_\theta(\hat{s},a_1)$. However, $\mathcal{Q}_\theta(s_1,a_1)>\mathcal{Q}_\theta(s_1,a_2)$. Furthermore, $\|s_1-\hat{s}\|=\frac{\epsilon}{2+\epsilon}<\epsilon$. Thus, $\mathcal{Q}_\theta$ is not ϵ-robust.

However, if we instead choose new parameters θ' for the state-action value function so that $\mathcal{Q}_{\theta'}(s_1,a_1)=0.8$ and $\mathcal{Q}_{\theta'}(s_1,a_2)=0.7$ one can easily check that $\mathcal{Q}_{\theta'}$ is ϵ-robust for all $\epsilon<0.1$.

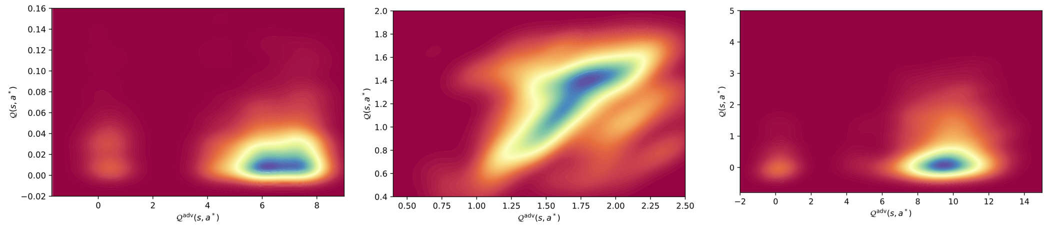

Figure 3: $\mathcal{Q}$ values of $\arg\max_{a\in\mathcal{A}} \mathcal{Q}(s,a)$ for adversarially and vanilla trained deep neural policies.

Furthermore, observe that $\mathcal{Q}_{\theta'}$ gives the correct ranking of actions in state s_1, but overestimates the optimal state-action value by a factor of $8/\epsilon$. $\square$

Next we demonstrate that this is a general phenomenon which occurs with neural-network approximation of the $\mathcal{Q}$-function in robust, i.e. adversarially, trained deep reinforcement learning policies.

4 Emprical Analysis in High-Dimensional MDPs

The empirical analysis is conducted in high dimensional state representation MDPs. In particular, our experiments are conducted in the Arcade Learning Environment (ALE) (Bellemare et al., 2013). The vanilla trained deep neural policy is trained via Double Deep Q-Network (DDQN) (Wang et al., 2016) initially proposed in (Hasselt et al., 2016a) with prioritized experience replay proposed by (Schaul et al., 2016), and the state-of-the-art adversarially trained deep neural policy is trained via State-Adversarial Double Deep Q-Network (SA-DDQN) (Section 2) with prioritized experience replay (Schaul et al., 2016). The results are averaged over 10 episodes. We explain in detail all the necessary hyperparameters for the implementation in the supplementary material. The standard error of the mean is included for all of the figures and tables. Note that in the main body of the paper we focus on the baseline adversarial training method. In the supplementary material we also provide analysis on the follow-up more recent studies in adversarial training techniques. The results reported for all of the adversarial training techniques remain the same: that the adversarially trained policies learn inaccurate, inconsistent and overestimated state-action values. Performance drop $\mathcal{P}$ is given by $\mathcal{P} = (\text{Score}_{\text{base}} - \text{Score}_{\text{actmod}})/(\text{Score}_{\text{base}} - \text{Score}_{\text{min}})$, where $\text{Score}_{\text{base}}$ represent the baseline run of the game with no action modification, $\text{Score}_{\text{min}}$ represents the minimum score available for a given game, and $\text{Score}_{\text{actmod}}$ represents the run of the game where the actions of the agent are modified for a fraction of the state observations. To measure the accuracy for the state-action value estimates formally, let a_i be the i^{th} best action decided by the deep neural policy in a given state s (i.e. $\mathcal{Q}(s,a)$ is sorted in decreasing order, and a_i is the action corresponding to i^{th} largest $\mathcal{Q}$-value). For a trained agent, the value of $\mathcal{Q}(s,a_i)$ should represent the expected cumulative rewards obtained by taking action a_i in state s, and then taking the highest $\mathcal{Q}$-value action (i.e. a_1) in every subsequent state. Thus, a natural test to perform would be: for a random state s the policy should take action a_i in state s, and the highest $\mathcal{Q}$-value action for the rest of the states. By comparing the relative performance drop $\mathcal{P}$ in this test to a clean run where the agent always takes the highest $\mathcal{Q}$-value action, one can measure the decline in rewards caused by taking action a_i. Further, we can provide a measure of accuracy for the state-action value function by comparing the results of the test for each $i \in \{1, 2 \dots |A|\}$, and checking that the relative performance drops $\mathcal{P}_i$ are in the correct order i.e. $0 = \mathcal{P}_1 \leq \mathcal{P}_2 \cdots \leq \mathcal{P}_{|A|}$. We take this one step further and analyze the performance drop with Ω-fraction of the states in the episode uniformly at random, and making the policy execute action a_i in each of the sampled states. We then record the relative performance drop as a function of Ω, yielding a performance drop curve $\mathcal{P}_i(\Omega)$. More formally, we define

Definition 4.1 (*Performance Drop Curve*)**.** Let $\mathcal{M}$ be an MDP and $\mathcal{Q}(s,a)$ be a state-action value function for $\mathcal{M}$. In each state label the actions $a_1, \dots a_{|A|}$ in order so that $\mathcal{Q}(s,a_1) \geq \mathcal{Q}(s,a_2) \cdots \geq \mathcal{Q}(s,a_{|A|})$. The *performance drop curve* $\mathcal{P}_i(\Omega)$ is the expected performance drop of an agent in $\mathcal{M}$ which takes action a_i in a randomly sampled Ω-fraction of states, and executes a_1 in all other states.

Using these performance drop curves one can confirm whether $\mathcal{P}_i(\Omega)$ lies above $\mathcal{P}_j(\Omega)$ whenever $i > j$. Yet to be precise we will quantify the relative ordering of the performance drop curves.

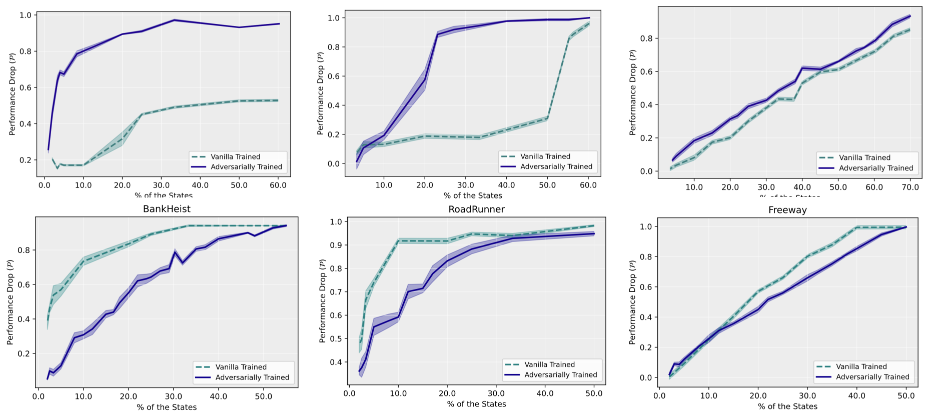

Figure 4: Up: Performance drop $\mathcal{P}_2(\Omega)$ with respect to action modification a_2 for the state-of-the-art certified (i.e. adversarially) and vanilla trained deep neural policies. Down: Performance drop $\mathcal{P}_w(\Omega)$ with respect to action modification a_w. Left: BankHeist. Center: RoadRunner. Right: Freeway.

Table 1: Area under the curve of performance drop under action modification (AM) a_2 and a_w for the state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies.

Environments	BankHeist		RoadRunner		Freeway	
Training Method	Adversarial	Vanilla	Adversarial	Vanilla	Adversarial	Vanilla
AM a_2	0.449±0.007	0.191±0.04	0.414±0.015	0.247±0.009	0.351±0.009	0.302±0.007
AM a_w	0.311± 0.011	0.398±0.011	0.345±0.011	0.393±0.009	0.241±0.007	0.311±0.010

Definition 4.2 (τ-*domination*)**.** Let $\mathcal{F} : [0, 1] \to [0, 1]$ and $\mathcal{G} : [0, 1] \to [0, 1]$. For any $\tau > 0$, we say that the $\mathcal{F}$ τ-dominates $\mathcal{G}$ if $\int_0^1 (\mathcal{F}(\Omega) - \mathcal{G}(\Omega))\, d\Omega > \tau$.

To compare the accuracy of state-action values for vanilla versus adversarially trained agents, we can thus perform the above test, and check the relative ordering of the curves $\mathcal{P}_i(\Omega)$ using Definition 4.2 for each agent type. In addition, we can also directly compare for each i the curve $\mathcal{P}_i^{\text{adv}}(\Omega)$ for the adversarially trained agent with the curve $\mathcal{P}_i^{\text{vanilla}}(\Omega)$ for the vanilla trained agent. This is possible because $\mathcal{P}_i(\Omega)$ measures the performance drop of the agent relative to a clean run, and so always takes values on a normalized scale from 0 to 1. Thus, if we observe for example that $\mathcal{P}_2^{\text{adv}}(\Omega)$ τ-dominates $\mathcal{P}_2^{\text{vanilla}}(\Omega)$ for some $\tau > 0$, we can conclude that the state-action value function of the vanilla trained agent more accurately represents the second-best action than that of the adversarially trained agent.

4.1 Randomized Decisions of Robust Reinforcement Learning

Figure 4 reports the performance drop $\mathcal{P}_2(\Omega)$ and $\mathcal{P}_w(\Omega)$ as a function of the fraction of states Ω in which the action modification is applied for certified trained deep neural policies and vanilla trained deep neural policies. In particular, the action modification is set for the second best action a_2 decided by the state-action value function $\mathcal{Q}(s, a)$. As we increase the fraction of states in which the action modification set to a_2 is applied, we observe a performance drop for both of the deep neural policies. However, we observe that the vanilla trained deep neural policies experience a lower performance drop with this modification. Especially in BankHeist we observe that the performance drop does not exceed 0.55 even when the action modification is applied for a large fraction of the visited states for the vanilla trained deep neural policies. This gap in the performance drop between the adversarially trained and vanilla trained deep neural policies indicates that the state-action value function learnt by vanilla trained deep neural policies has a better estimate for the state-action values. As we measured the impact of a_2 modification on the policy performance, we further test $a_w = \arg\min_a \mathcal{Q}(s, a)$ (i.e. worst possible action in a given state) modification on the deep neural policy. Figure 4 shows that the performance drop $\mathcal{P}_w(\Omega)$ is higher in the vanilla trained deep neural policies compared to adversarially trained deep neural policies when the action modification is set to a_w. This again further demonstrates that the state-action value function learnt by the vanilla trained deep neural policy has a more accurate representation. We argue that adversarial training places higher emphasis on ensuring that the highest ranked action (i.e. the action that maximizes the state-action value function in a given state) does not change under small ℓ_p-norm bounded perturbations, rather than accurately computing the state-action value function as discussed in Section 3. A method which places higher emphasis on the highest ranked action risks converging to a state-action value function with overestimated $\mathcal{Q}$-values. We further demonstrate this in Section 4.3.

4.2 Counterfactual Decisions and The Misalignment of Adversarial Training

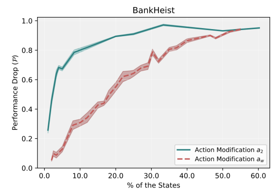

Figure 5: $\mathcal{P}_2$ and $\mathcal{P}_w$ of adversarial training.

The results reported in this section demonstrate the misalignment between deep neural policies and human decision making caused by robust training. Reinforcement learning is founded on the inspiration drawn from natural intelligence (Watkins, 1989; Kehoe et al., 1987; Romo & Schultz, 1990; Montague et al., 1996) providing further theoretical guarantees on its limitations and capabilities (Watkins & Dayan, 1992; Sutton, 1988; Barto et al., 1995). Our analysis and results demonstrate that an extensive recent line of work myopically focusing on safety diverts the main contributions and the tight core connection of reinforcement learning with neuroscience while producing policies that are both in fact not safe and misaligned. In particular, Figure 5 demonstrates that choosing the worst action leads to a smaller performance drop than choosing the second best action i.e. $\mathcal{P}_w(\Omega) < \mathcal{P}_2(\Omega)$ for all Ω in BankHeist. Notably, the results reported in Figure 5 reveal that robust training methods assign random values to the counterfactual actions which is a direct misalignment with natural intelligence decision making. The results reported in Figure 4 demonstrate the clear juxtaposition between standard reinforcement learning and safety concerned reinforcement learning, i.e. robust trained. Intriguingly, these results reveal that standard reinforcement learning indeed learns aligned values with natural intelligence; however, robust training converts these values to be misaligned. Furthermore, the misalignment of the adversarial, i.e. robust, training causes these deep neural policies to learn inconsistent action ranking which can be seen as a vulnerability problem from a security point of view. Nonetheless, most intriguingly these results demonstrate the foundational loss of information in the state-action value function as a novel fundamental trade-off intrinsic to adversarial training.

4.3 Overestimation of $\mathcal{Q}$-values in Adversarially Trained Deep Neural Policies

Overestimation of $\mathcal{Q}$-values was initially discussed by Thrun & Schwartz (1993) as a byproduct of the use of function approximators, and was subsequently explained as being caused by the use of the max operator in approximating the maximum of the expected $\mathcal{Q}$-values (van Hasselt, 2010). Furthermore, it has been shown that the overestimation bias results in learning sub-optimal policies (Hasselt et al., 2016a), and thus the deep double-$\mathcal{Q}$ learning algorithm has been proposed to alleviate the overestimation problem (Hasselt et al., 2016a), that was initially observed in DQN (Mnih et al., 2016). In this section we empirically demonstrate that state-of-the-art certified training indeed leads to overestimation in $\mathcal{Q}$-values, as has been theoretically predicted in Section 3. In particular, Figure 3 reports the overestimation bias on the state-action values learned by the adversarially trained deep neural policies. Note that the fact that adversarially trained deep reinforcement learning policies assign higher state-action values than the vanilla trained deep reinforcement learning policies while performing similarly, i.e. obtaining similar expected cumulative rewards, clearly demonstrates that the adversarial training techniques, on top of the inconsonance and the inaccuracy issues, learn explicitly biased state-action values.

While these state-of-the-art adversarial training algorithms have attracted a significant level of attention from the research community, i.e. multiple spotlight presentations in NeurIPS, to encourage more efforts on this line of research to ensure that these policies will not cause harm and benefit humanity, it carries a significant level of responsibility to reveal the principal vulnerabilities of these models. The uncovered issues with this line of algorithms carry utmost importance due to the fact that these studies influence future research directions while significantly pivoting research focus. Furthermore, without the knowledge of the actual costs and drawbacks of these algorithms a significant level of research efforts might be misdirected. While the results reported in Figure 5, Section 4.1, and Section 4.3 reveal concrete problems of the state-of-the-art adversarial training techniques particularly regarding the inconsonance and overestimation issues, from the security perspective these results call for an urgent reconsideration and discussion on the certified robustness algorithms and their implications.

Table 2: Normalized state-action value estimates and state-action value estimate shift for the second best action for certified adversarially trained and vanilla trained deep reinforcement learning policies.

$\mathcal{Q}$ Estimates	$\mathcal{Q}(s,a^*)$		$\mathcal{Q}(s,a_2)$		$\mathcal{Q}(s,a_w)$	
ALE	Adversarial	Vanilla	Adversarial	Vanilla	Adversarial	Vanilla
BankHeist	0.1894±0.002	0.170±0.003	0.130±0.0006	0.169±0.002	0.127±0.0010	0.161±0.004
RoadRunner	0.1696±0.008	0.236±0.094	0.132±0.0026	0.159±0.079	0.126±0.0049	-0.265±0.071
Freeway	0.1894±0.002	0.341±0.008	0.130±0.0006	0.333±0.002	0.127±0.0010	0.325±0.009

4.4 Action Gap Phenomenon

The action gap is defined as the difference $\mathcal{Q}$-values

$$\mathcal{G}(\mathcal{Q},s) = \max_{\hat{a}\in\mathcal{A}} \mathcal{Q}(s,\hat{a}) - \max_{a\in\mathcal{A}_s^{\perp}} \mathcal{Q}(s,a).$$

A connection between the action gap and the approximation errors has been mentioned in prior studies (Bellemare et al., 2016) and have been hypothesized that increasing the action gap of the learned value function causes a decrease in overestimation of $\mathcal{Q}$-values. Following this study, several papers built on the hypothesis that increasing the action gap causes reduction in bias. However, our results reveal that targeting to increase the action gap must be upper-bounded by the preserving the order of the counterfactual actions to obtain truly robust and safe policies. Once this upperbound is passed the policy forms values that are misaligned with human decision making. To preserve the initial core foundations of reinforcement learning and its alignment with human decision making process we must preserve the approaches that targeted learning methods align and matched natural intelligence decision making (Baird & Moore, 1993; Watkins & Dayan, 1992; Averbeck & Costa, 2017; Wang et al., 2018).

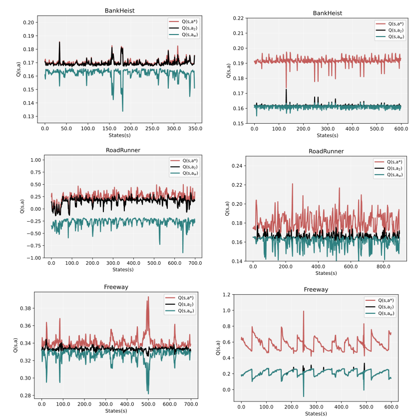

Figure 6: Normalized state-action values for the best action a^*, second best action a_2 and worst action a_w over states. Left: Vanilla trained. Right: State-of-the-art adversarially trained[2].

5 Conclusion

In this paper we focus on the juxtaposition of human decision making and reinforcement learning within the realm of alignment of robust training. We provide an extensive theoretical analysis on the on the fundamental effects of robust training compared to standard reinforcement learning. Both our empirical analysis conducted in high-dimensional state representation MDPs and theoretical analysis demonstrate that standard deep reinforcement learning is aligned with the human decision making process while techniques focused on providing certified safety and robustness are in fact misaligned. More intriguingly, we demonstrate that this misalignment reaches up to a level that adversarially, i.e. robust, trained deep neural policies completely lose all the information in the state-action value function that contains the relative ranking of the actions. Moreover, orthogonal to misalignment issues our theoretical analysis reveals the fundamental trade-off in robust training methods. Our results demonstrate that the *certified-safety* claims of the prior line of research fail to deliver their promises, and our paper discovers manifold issues with certified training regarding what truly robust training methods learn. Our investigation while highlighting the gap between natural intelligence decision making and certified training, further lays out the intrinsic properties of adversarial training while systematically revealing the underlying vulnerabilities, and thence can be conducive to building truly robust and aligned deep neural policies.

[2]Figure 6 reports that robust, i.e. adversarial, training increases the action gap, yet still learns overestimated state-action values. See supplementary material for further discussion on the action gap and the connection we highlight between consistent Bellman operator and the implicit Kullback-Leibler regularization. Note that due to the fact that the adversarially trained deep neural policy overestimates $\mathcal{Q}$-values, we introduce a normalization in order to compare the action gaps of adversarially and vanilla trained policies. In particular, in Figure 6 we report normalized $\mathcal{Q}$-values in each state s by dividing $\mathcal{Q}(s,a)$ by $\sum_a |\mathcal{Q}(s,a)|$.

References

Bruno B Averbeck and Vincent D Costa. Motivational neural circuits underlying reinforcement learning. In *Nature Neuroscience*, 2017.

Leemon Baird and Andrew Moore. Reinforcement learning through gradient descent. In *Conference on Neural Information Processing Systems, NeurIPS*, 1993.

Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh. Learning to act using real-time dynamic programming. In *Artificial Intelligence*, 1995.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, pp. 253–279, 2013.

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.

John M Danskin. A method for solving a convex programming problem with convergence rate $o(1/k^2)$. New York: Springer, 1967.

Adam Gleave, Michael Dennis, Cody Wild, Kant Neel, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *International Conference on Learning Representations ICLR*, 2020.

Ian Goodfellow, Jonathan Shelens, and Christian Szegedy. Explaning and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Fabian Grabenhorst and Edmund Rolls. Value, pleasure and choice in the ventral prefrontal cortex. *Trends in Cognitive Sciences*, 2011.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016a.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *In Thirtieth AAAI conference on artificial intelligence*, 2016b.

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

Zhang Huan, Chen Hongge, Xiao Chaowei, Bo Li, Mingyan Boning, Duane Liu, and ChoJui Hsiesh. Robust deep reinforcement learning against adversarial perturbations on state observatons. *NeurIPS Spotlight Presentation*, 2020.

Sandy Huang, Nicholas Papernot, Yan Goodfellow, Ian an Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *Workshop Track of the 5th International Conference on Learning Representations*, 2017.

Steven Kapturowski, Victor Campos, Ray Jiang, Nemanja Rakicevic, Hado van Hasselt, Charles Blundell, and Adrià Puigdomènech Badia. Human-level atari 200x faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023.

E. James Kehoe, Bernard G. Schreurs, and Peita Graham. Temporal primacy overrides prior training in serial compound conditioning of the rabbit's nictitating membrane response. Animal Learning and Behavior, 1987.

Miriam Klein-Flügge, Steven W. Kennerley, Karl Friston, and Sven Bestmann. Neural signatures of value comparison in human cingulate cortex during decisions requiring an effort-reward trade-off. In *The Journal of Neuroscience*, 2016.

Ezgi Korkmaz. Nesterov momentum adversarial perturbations in the deep reinforcement learning domain. *International Conference on Machine Learning, ICML 2020, Inductive Biases, Invariances and Generalization in Reinforcement Learning Workshop.*, 2020.

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

Ezgi Korkmaz. Understanding and Diagnosing Deep Reinforcement Learning Decision Making. In *International Conference on Machine Learning, ICML 2024*, 2024.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *International Conference on Learning Representations*, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Brian Lau and Paul W. Glimcher. Action and outcome encoding in the primate caudate nucleus. In *Journal of Neuroscience*, volume 27, pp. 14502–14514, 2007.

Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. The Annual Review of Neuroscience, 2012.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, arc G Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533, 2015.

Volodymyr Mnih, Adria Badia Puigdomenech, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *In International Conference on Machine Learning*, pp. 1928–1937, 2016.

Read Montague, Peter Dayan, and Terrence Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of Neuroscience, 1996.

Wei-Xing Pan, Robert Schmidt, Jeffery R. Wickens, and Brian I. Hyland. Dopamine cells respond to predicted events during classical conditioning: Evidence for eligibility traces in the reward-learning network. Journal of Neuroscience, 2005.

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

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *International Conference on Learning Representations ICLR*, 2017.

Ranulfo Romo and Wolfram Schultz. Dopamine neurons of the monkey midbrain: contingencies of responses to active touch during self-initiated arm movements. Journal of Neurophysiology, 1990.

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

Wolfram Schultz, Paul Apicella, and Tomas Ljungberg. Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task. Journal of Neuroscience, 1993.

Richard Sutton. Temporal credit assignment in reinforcement learning. PhD Thesis University of Massachusetts Amherst, 1984.

Richard Sutton. Learning to predict by the methods of temporal difference. Machine Learning, 1988.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dimutru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. *In Fourth Connectionist Models Summer School*, 1993.

Hado van Hasselt. Double q-learning. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta (eds.), *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pp. 2613–2621. Curran Associates, Inc., 2010.

Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. In *Nature Neuroscience*, 2018.

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.

Chris Watkins. Learning from delayed rewards. In *PhD thesis, Cambridge*. King's College, 1989.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. 1992.

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.