Deep Reinforcement Learning Policies Learn Shared Adversarial Features across MDPs

Ezgi Korkmaz

Abstract

The use of deep neural networks as function approximators has led to striking progress for reinforcement learning algorithms and applications. Yet the knowledge we have on decision boundary geometry and the loss landscape of neural policies is still quite limited. In this paper, we propose a framework to investigate the decision boundary and loss landscape similarities across states and across MDPs. We conduct experiments in various games from Arcade Learning Environment, and discover that high sensitivity directions for neural policies are correlated across MDPs. We argue that these high sensitivity directions support the hypothesis that non-robust features are shared across training environments of reinforcement learning agents. We believe our results reveal fundamental properties of the environments used in deep reinforcement learning training, and represent a tangible step towards building robust and reliable deep reinforcement learning agents.

1 Introduction

Building on the success of DNNs for image classification, deep reinforcement learning has seen remarkable advances in various complex environments Mnih et al. (2015); Schulman et al. (2017); Lillicrap et al. (2016). Along with these successes come new challenges stemming from the current lack of understanding of the structure of the decision boundary and loss landscape of neural network policies. Notably, it has been shown that the high sensitivity of DNN image classifiers to imperceptible perturbations to inputs also occurs for neural policies Huang et al. (2017); Lin et al. (2017); Korkmaz (2020). This lack of robustness is especially critical for deep reinforcement learning, where the actions taken by the agent can have serious real-life consequences Levin and Carrie (2018).

Recent work has shown that adversarial examples are a consequence of the existence of non-robust (i.e. adversarial) features of datasets used in image classifier training Ilyas et al. (2019). That is, there are certain features which are actually useful in classifying the data, but are extremely sensitive to small perturbations, and thus incomprehensible to humans.

The existence in standard datasets of non-robust features, which both generalize well but are simultaneously highly sensitive to small perturbations, raise serious concerns about the way that DNN models are currently trained and tested. On the one hand, since non-robust features are actually useful in classification, current training methods which optimize for classification accuracy have no reason to ignore them. On the other hand, since these features can be altered with visually imperceptible perturbations they present a formidable obstacle to constructing models that behave anything like humans. Instead, models trained in the presence of non-robust features are likely to have directions of high-sensitivity (or equivalently small margin) correlated with these features.

In the reinforcement learning setting, where policies are generally trained in simulated environments, the presence of non-robust features leading to high-sensitivity directions for neural network policies raises serious concerns about the ability of these policies to generalize beyond their training simulations. Consequently, identifying the presence of non-robust features in simulated training environments and understanding their effects on neural-network policies is an important first step in designing algorithms for training reinforcement learning agents which can perform well in real-world settings.

In this paper we study the decision boundaries of neural network policies in deep reinforcement learning. In particular, we ask: how are high-sensitivity directions related to the actions taken across states for a neural network policy? Are high-sensitivity directions correlated between neural policies trained in different MDPs? Do state-of-the-art adversarially trained deep reinforcement learning policies inherit similar high-sensitivity directions to vanilla trained deep reinforcement learning policies? Do non-robust features exist in the deep reinforcement learning training environments? To answer these questions, we propose a framework based on identifying directions of low distance to the neural policy decision boundary, and investigating how these directions affect the decisions of the agents across states and MDPs. Our main contributions are as follows:

- We introduce a framework based on computing a high-sensitivity direction in one state, and probing the decision boundary of the neural policy along this direction as the neural policy is executed in a set of carefully controlled scenarios.

- We examine the change in the action distribution of an agent whose input is shifted along a high-sensitivity direction, and show that in several cases these directions correspond to shifts towards the same action across different states. These results lend credence to the hypothesis that high-sensitivity directions for neural policies correspond to non-robust discriminative features.
- We investigate the state-of-the-art adversarially trained deep reinforcement learning policies and show that adversarially trained deep reinforcement learning policies share high sensitivity directions with vanilla trained deep reinforcement learning policies.
- Via experiments in the Arcade Learning Environment we rigorously show that the high-sensitivity directions computed in our framework correlate strongly across states and in several cases across MDPs. This suggests that distinct MDPs from standard baseline environments contain correlated non-robust features that are utilized by deep neural policies.

2 Related Work and Background

2.1 Deep Reinforcement Learning

In this paper we examine discrete action space MDPs that are represented by a tuple: $\mathcal{M} = (S, A, P, r, \gamma, s_0)$ where S is a set of states, A is a set of discrete actions, $P : S \times A \times S \to \mathbb{R}$ is the transition probability, $r : S \times A \to \mathbb{R}$ is the reward function, γ is the discount factor, and s_0 is the initial state distribution. The agent interacts with the environment by observing $s \in S$, taking actions $a \in A$, and receiving rewards $r : S \times A \to \mathbb{R}$. A policy $\pi : S \times A \to \mathbb{R}$ assigns a probability distribution over actions $\pi(s, \cdot)$ to each state s. The goal in reinforcement learning is to learn a policy π that maximizes the expected cumulative discounted reward $R = \mathbb{E}[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)]$ where $a_t \sim \pi(s_t, \cdot)$. For an MDP $\mathcal{M}$ and policy π we call a sequence of state, action, reward, next state tuples, (s_i, a_i, r_i, s'_i), that occurs when utilizing π in $\mathcal{M}$ an episode. We use $p_{\mathcal{M},\pi}$ to denote the probability distribution over the episodes generated by the randomness in $\mathcal{M}$ and the policy π. In Q-learning the goal of maximizing the expected discounted cumulative rewards is achieved by building a state-action value function $Q(s, a) = \mathbb{E}_{s \sim \pi(s,\cdot)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\right]$. The function $Q(s, a)$ is intended to represent the expected cumulative rewards obtained by taking action a in state s, and in all future states s' taking the action a' which maximizes $Q(s', a')$.

2.2 Adversarial Perturbation Methods

In order to identify high-sensitivity directions for neural policies we use methods designed to compute adversarial perturbations of minimal ℓ_p-norm. The first paper to discuss adversarial examples for DNNs was Szegedy et al. (2014). Subsequently Goodfellow, Shelens, and Szegedy (2015) introduced the fast gradient sign method (FGSM) which computes adversarial perturbations by maximizing the linearization of the cost function in the ℓ_∞-ball of radius ϵ.

$$x_{\text{adv}} = x + \epsilon \cdot \frac{\nabla_x J(x, y)}{\|\nabla_x J(x, y)\|_p}, \quad (1)$$

Here x is the clean input, y is the correct label for x, and J is the cost function used to train the neural network classifier. Further improvements were obtained by Kurakin, Goodfellow, and Bengio (2017) by using FGSM gradients to iteratively search within the ℓ_p-norm ball of radius ϵ.

Currently, the state-of-the-art method for computing minimal adversarial perturbations is the formulation proposed by Carlini and Wagner (2017). In particular, Athalye, Carlini, and Wagner (2018) and Tramer et al. (2020) showed that the C&W formulation can overcome the majority of the proposed defense and detection algorithms. For this reason, in this work we create the adversarial perturbations mostly by utilizing the C&W formulation and its variants. In particular, in the deep reinforcement learning setup the C&W formulation is,

$$\min_{s_{\text{adv}} \in S} c \cdot J(s_{\text{adv}}) + \|s_{\text{adv}} - s\|_2^2 \quad (2)$$

where s is the unperturbed input, s_{adv} is the adversarially perturbed input, and $J(s)$ is the augmented cost function used to train the network. Note that in the deep reinforcement learning setup the C&W formulation finds the minimum distance to a decision boundary in which the action taken by the deep reinforcement learning policy is non-optimal. The second method we use to produce adversarial examples is the ENR method Chen et al. (2018),

$$\min_{s_{\text{adv}} \in S} c \cdot J(s_{\text{adv}}) + \lambda_1 \|s_{\text{adv}} - s\|_1 + \lambda_2 \|s_{\text{adv}} - s\|_2^2 \quad (3)$$

By adding ENR this method produces sparser perturbations compared to the C&W formulation with similar ℓ_2-norm.

2.3 Deep Reinforcement Learning and Adversarial Perspective

The first work on adversarial examples in deep reinforcement learning appeared in Huang et al. (2017) and Kos and Song (2017). These two concurrent papers demonstrated that FGSM adversarial examples could significantly decrease the performance of deep reinforcement learning policies in standard baselines. Follow up work by Pattanaik et al. (2018) introduced an alternative to the standard FGSM by utilizing an objective function which attempts to increase the probability of the worst possible action (i.e. the action a_w minimizing $Q(s, a)$). The work of Lin et al. (2017) and Sun et al. (2020) attempts to exploit the difference in the importance of states across time by only introducing adversarial perturbations at strategically chosen moments. In these papers the perturbations are computed via the C&W formulation. Korkmaz (2021c) focuses on exposing vulnerabilities of deep reinforcement learning policies in both spatial and frequency domain. Due to the fragilities outlined in the papers mentioned above, there has been another line of work on attempting to train agents that are robust to adversarial perturbations. In this line of work Pinto et al. (2017) proposed modelling the agent and the adversary as playing a zero-sum game, and training both agent and adversary simultaneously. A similar idea of modelling the agent vs the adversary appears in Gleave et al. (2020), but with an alternative constraint that the adversary must take natural actions in the simulated environment rather than introducing ℓ_p-norm bounded perturbations to the agent's observations. Most recently, Zhang et al. (2020) introduced the notion of a State-Adversarial MDP, and used this new definition to design an adversarial training algorithm for deep reinforcement learning with more solid theoretical motivation. Subsequently, several concerns have been raised regarding the problems introduced by adversarial training (Korkmaz 2021a,b,c).

3 High-Sensitivity Directions

The approach in our paper is based on identifying directions of high sensitivity for neural policies. A first attempt to define the notion of high-sensitivity direction might be to say that $\boldsymbol{v}$ is a high sensitivity direction for π at state s if for some small $\epsilon > 0$

$$\arg\max_a \pi(s + \epsilon \boldsymbol{v}, a) \neq \arg\max_a \pi(s, a)$$

but for a random direction $\boldsymbol{r} = \frac{\|\boldsymbol{v}\|}{\|\boldsymbol{g}\|}\boldsymbol{g}$ with $\boldsymbol{g} \sim \mathcal{N}(0, I)$

$$\arg\max_a \pi(s + \epsilon \boldsymbol{r}, a) = \arg\max_a \pi(s, a)$$

with high probability over the random vector $\boldsymbol{r}$. In other words, $\boldsymbol{v}$ is a high sensitivity direction if small perturbations along $\boldsymbol{v}$ change the action taken by policy π in state s, but perturbations of the same magnitude in a random direction do not. There is a subtle issue with this definition however.

In reinforcement learning there are no ground truth labels for the correct action for an agent to take in a given state. While the above definition requires that the policy switches which action is taken when the input is shifted a small amount in the $\boldsymbol{v}$ direction, there is no *a priori* reason that this shift will be bad for the agent.

Instead, the only objective metric we have of the performance of a policy is the final reward obtained by the agent. Therefore we define high-sensitivity direction as follows:

Definition 3.1. Recall that $R = \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$ is the cumulative reward. A vector $\boldsymbol{v}$ is a *high-sensitivity direction* for a policy π if there is an $\epsilon > 0$ such that

$$\mathbb{E}_{a_t \sim \pi(s_t + \epsilon \boldsymbol{v}, \cdot)}[R] \ll \mathbb{E}_{a_t \sim \pi(s_t, \cdot)}[R]$$

but for a random direction $\boldsymbol{r} = \frac{\|\boldsymbol{v}\|}{\|\boldsymbol{g}\|}\boldsymbol{g}$ with $\boldsymbol{g} \sim \mathcal{N}(0, I)$

$$\mathbb{E}_{a_t \sim \pi(s_t + \epsilon \boldsymbol{r}, \cdot)}[R] \approx \mathbb{E}_{a_t \sim \pi(s_t, \cdot)}[R].$$

In short, $\boldsymbol{v}$ is a high-sensitivity direction if small perturbations along v significantly reduce the expected cumulative rewards of the agent, but the same magnitude perturbations along random directions do not. This definition ensures not only that small perturbations in the direction $\boldsymbol{v}$ cross the decision boundary of the neural policy in many states s, but also that the change in policy induced by these perturbations has a semantically meaningful impact on the agent.

To see how high-sensitivity directions arise naturally consider the linear setting where we think of s as a vector of features in $\mathbb{R}^n$, and for each action a we associate a weight vector $\boldsymbol{w}_a \in \mathbb{R}^n$. The policy in this setting is given by deterministically taking the action $\arg\max_{a \in A}\langle \boldsymbol{w}_a, s\rangle$ in state s. We assume that the weight vectors are not too correlated with each other: $\langle \boldsymbol{w}_a, \boldsymbol{w}_{a'}\rangle < \alpha \cdot \min\{\|\boldsymbol{w}_a\|^2, \|\boldsymbol{w}_{a'}\|^2\}$ for some constant $\alpha < 1$. We also assume that the lengths of the weight vectors are not too different: for all a, a' $\|\boldsymbol{w}_a\| \leq \beta \|\boldsymbol{w}_{a'}\|$ for some constant $\beta > 1$.

Suppose that there is a set of states S_1 accounting for a significant fraction of the total rewards such that: (1) by taking the optimal action $a^*(s)$ for $s \in S_1$ the agent receives reward 1, and (2) there is one action b, such that taking action b gives reward 0 for a significant fraction of states in S_1. We claim that, assuming that there is a constant gap between $\langle \boldsymbol{w}_{a^*(s)}, s\rangle$ and $\langle \boldsymbol{w}_a, s\rangle$ for $a \neq a^*(s)$ then $\boldsymbol{v} = \boldsymbol{w}_b$ is a high-sensitivity direction.

Proposition 3.2. *Assume that there exist constants $c, d > 0$ such that $c < \langle \boldsymbol{w}_{a^*(s)}, s\rangle - \langle \boldsymbol{w}_a, s\rangle < d$ for all $a \neq a^*(s)$ and $s \in S_1$. Then $\boldsymbol{w}_b$ is a high-sensitivity direction.*

Proof. See full version of the paper. □

The empirical results in the rest of the paper confirm that high-sensitivity directions do occur in deep neural policies, and further explore the correlations between these directions across states and across MDPs.

4 Framework for Investigating High-Sensitivity Directions

In this section we introduce a framework to seek answers for the following questions:

- *Are high sensitivity directions shared amongst states in the same MDP?*
- *Is there a correlation between high sensitivity directions across MDPs and across algorithms?*
- *Do non-robust features exist in the deep reinforcement learning training environments?*

It is important to note that the goal of this framework is not to demonstrate the already well-known fact that adversarial perturbations are a problem for deep reinforcement learning. Rather, we are interested in determining whether high-sensitivity directions are correlated across states, across MDPs and across training algorithms. The presence of such correlation would indicate that non-robust features are an intrinsic property of the training environments themselves. Thus, understanding the extent to which non-robust features correlate in these ways can serve as a guide for how we should design algorithms and training environments to improve robustness.

Given a policy π and state s we compute a direction $\boldsymbol{v}(s, \pi(s, \cdot))$ of minimal margin to the decision boundary of $\pi(s, \cdot)$ in state s. We fix a bound κ such that perturbations of norm κ in a random direction $\boldsymbol{r}$ with $\|\boldsymbol{r}\|_2 = 1$ have insignificant impact on the rewards of the agent.

$$\mathbb{E}_{a_t \sim \pi(s_t + \kappa \boldsymbol{r}, \cdot)}[R] \approx \mathbb{E}_{a_t \sim \pi(s_t, \cdot)}[R].$$

Our framework is based on evaluating the sensitivity of a neural policy along the direction $\boldsymbol{v}$ in a sequence of increasingly general settings. Each setting is defined by (1) the state s chosen to compute $\boldsymbol{v}(s, \pi(s, \cdot))$, and (2) the states s' where a small perturbation along $\boldsymbol{v}(s, \pi(s, \cdot))$ is applied when determining the action taken.

Definition 4.1. *Individual state setting*, $\mathcal{A}^{\text{individual}}$, is the setting where in each state s a new direction is computed and a perturbation along that direction is applied. In each state s_i we compute

$$s_i^* = s_i + \kappa \cdot \frac{\boldsymbol{v}(s_i, \pi(s_i, \cdot))}{\|\boldsymbol{v}(s_i, \pi(s_i, \cdot))\|_2} \tag{4}$$

and then take an action determined by $\pi(s_i^*, \cdot)$.

The individual state setting acts as a baseline to which we can compare the expected rewards of an agent with inputs perturbed along a single direction $\boldsymbol{v}$. If the decline in rewards for an agent whose inputs are perturbed along a single direction is close to the decline in rewards for the individual state setting, this can be seen as evidence that the direction $\boldsymbol{v}$ satisfies the first condition of Definition 3.1.

Definition 4.2. *Episode independent random state setting*, $\mathcal{A}^{\text{random}}_{\text{e}}$, is the setting where a random state s is sampled from a random episode e, and the perturbation $\boldsymbol{v}(s, \pi(s, \cdot))$ is applied to all the states visited in another episode e'. Sample $e \sim p_{\mathcal{M}}$ and $s \sim e$. Given an episode $e' \sim p_{\mathcal{M}}$, in each state s_i' of e' we compute

$$s_i^* = s_i' + \kappa \cdot \frac{\boldsymbol{v}(s, \pi(s, \cdot))}{\|\boldsymbol{v}(s, \pi(s, \cdot))\|_2} \tag{5}$$

and then take an action determined by $\pi(s_i^*, \cdot)$.

The episode independent random state setting is designed to identify high-sensitivity directions $\boldsymbol{v}$. By comparing the return in this setting with the case of a random direction $\boldsymbol{r}$ as described in Definition 3.1 we can decide whether $\boldsymbol{v}$ is a high-sensitivity direction.

Definition 4.3. *Environment independent random state setting*, $\mathcal{A}^{\text{random}}_{\mathcal{M}}$, is the setting where a random state $s(\mathcal{M})$ is sampled from a random episode of the MDP $\mathcal{M}$, and the perturbation $\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))$ is applied to all the states visited in an episode of a different MDP $\mathcal{M}'$. Sample $e \sim \mathcal{M}$ and $s(\mathcal{M}) \sim e$. Given an episode $e' \sim p_{\mathcal{M}'}$, in each state s_i' we compute

$$s_i^* = s_i' + \kappa \cdot \frac{\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))}{\|\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot)))\|_2} \tag{6}$$

and then take an action determined by $\pi(s_i^*, \cdot)$.

The environment independent random state setting is designed to test whether high-sensitivity directions are shared across MDPs. As with the episode independent setting, comparing with perturbations in a random direction allows us to conclude whether a high-sensitivity direction for $\mathcal{M}$ is also a high-sensitivity direction for $\mathcal{M}'$.

Definition 4.4. *Algorithm independent random state setting*, $\mathcal{A}^{\text{random}}_{\text{alg}}$, is the setting where a random state $s(\mathcal{M})$ is sampled from a random episode of the MDP $\mathcal{M}$, and the perturbation $\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))$ is applied to all the states visited in an episode for a policy π' trained with a different algorithm. Sample $e \sim \mathcal{M}$ and $s(\mathcal{M}) \sim e$. Given an episode $e' \sim p_{\mathcal{M}'}$, in each state s_i' we compute

$$s_i^* = s_i' + \kappa \cdot \frac{\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))}{\|\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot)))\|_2} \tag{7}$$

and then take an action determined by $\pi'(s_i^*, \cdot)$.

Note that in the above definition we allow $\mathcal{M} = \mathcal{M}'$, which corresponds to transferring perturbations between training algorithms in the same MDP. We will refer to the setting where $\mathcal{M} \neq \mathcal{M}'$ (i.e. transferring between algorithms and between MDPs at the same time) as algorithm and environment independent, and denote this setting by $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$. Algorithm 1 gives the implementation for Definition 4.4.

Algorithm 1: High-sensitivity directions with $\mathcal{A}^{\text{random}}_{\text{alg}}$

```
Input: Episode $e$ of MDP $\mathcal{M}$, state $s(\mathcal{M}) \sim e$, policy $\pi(s, \cdot)$, policy $\pi'(s, \cdot)$, episode $e'$ of MDP $\mathcal{M}'$, perturbation bound $\kappa$
  for $s_i'$ in $e'$ do
      $s_i^* \leftarrow s_i' + \kappa \cdot \frac{\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))}{\|\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))\|_2}$
      $a^*(s_i^*) = \arg\max_a \pi'(s_i^*, a)$
  end for
  return $\mathbb{E}_{a^*(s_i^*)}[R]$
```

5 Experiments

The Arcade Learning Environment (ALE) is used as a standard baseline to compare and evaluate new deep reinforcement learning algorithms as they are developed (Hasselt, Guez, and Silver 2016; Mnih et al. 2016; Wang et al. 2016; Fedus et al. 2020; Rowland et al. 2019; Mnih et al. 2015; Hessel et al. 2018; Kapturowski et al. 2019; Dabney et al. 2018; Dabney, Ostrovski, and Barreto 2021; Xu et al. 2020; Schmitt, Hessel, and Simonyan 2020). As a result, any systematic issues with these environments are of critical importance. Any bias within the environment that favors some algorithms over others risks influencing the direction of research in deep reinforcement learning for the next several years. This influence could take the form of diverting research effort away from promising algorithms, or giving a false sense of security that certain algorithms will perform well under different conditions. Thus, for these reasons it is essential to investigate the existence of non-robust features and the correlations between high-sensitivity directions within the ALE.

In our experiments agents are trained with Double Deep Q-Network (DDQN) proposed by Wang et al. (2016) with prioritized experience replay Schaul et al. (2016) in the ALE introduced by Bellemare et al. (2013) with the OpenAI baselines version Brockman et al. (2016). The state-of-the-art adversarially trained deep reinforcement learning polices

Settings [Adversarial Technique]	BankHeist	RoadRunner	JamesBond	CrazyClimber	TimePilot	Pong
$\mathcal{A}^{\text{individual}}$ [ENR]	0.646±0.018	0.821± 0.046	0.098±0.047	0.750±0.030	0.815±0.049	0.995±0.003
$\mathcal{A}^{\text{individual}}$ [C&W]	0.694±0.045	0.876±0.017	0.038±0.050	0.646±0.056	0.334±0.107	1.0±0.000
$\mathcal{A}_{\text{e}}^{\text{random}}$[ENR]	0.764±0.022	0.961±0.008	0.612±0.040	0.980± 0.002	0.517±0.085	1.0±0.000
$\mathcal{A}_{\text{e}}^{\text{random}}$[C&W]	0.118±0.041	0.649±0.041	0.019±0.058	0.956±0.003	0.228±0.097	1.0±0.000
$\mathcal{A}_{\mathcal{M}}^{\text{random}}$[ENR]	0.496±0.055	0.816±0.034	0.910±0.051	0.993±0.002	0.312±0.118	0.946±0.017
$\mathcal{A}_{\mathcal{M}}^{\text{random}}$[C&W]	0.022±0.0349	0.919±0.018	0.304±0.038	0.036±0.028	0.017±0.073	0.966±0.008
Gaussian	0.078±0.037	0.027±0.025	0.038±0.063	0.054±0.025	0.031±0.063	0.045±0.018

Table 1: Impacts of C&W and ENR formulation for the proposed framework for investigating high sensitivity directions cross-states and cross-MDPs.

utilize the State-Adversarial DDQN (SA-DDQN) algorithm proposed by Zhang et al. (2020) with prioritized experience replay Schaul et al. (2016). We average over 10 episodes in our experiments. We report the results with standard error of the mean throughout the paper. The impact on an agent is defined by normalizing the performance drop as follows

$$\text{Impact} = \frac{\text{Score}_{\text{max}} - \text{Score}_{\text{set}}}{\text{Score}_{\text{max}} - \text{Score}_{\text{min}}}. \tag{8}$$

Here $\text{Score}_{\text{max}}$ is the score of the baseline trained agent following the learned policy in a clean run of the agent in the environment, $\text{Score}_{\text{set}}$ is the score of the agent with settings introduced in Section 4, and $\text{Score}_{\text{min}}$ is the score the agent receives when choosing the worst possible action in each state. All scores are recorded at the end of an episode.

5.1 Investigating High-Sensitivity Directions

Table 1 shows impact values in games from the Atari Baselines for the different settings from our framework utilizing the C&W formulation, ENR formulation and Gaussian noise respectively to compute the directions $\boldsymbol{v}$. In all the experiments we set the ℓ_2-norm bound κ in our framework to a level so that Gaussian noise with ℓ_2-norm κ has insignificant impact. In Table 1 we show the Gaussian noise impacts on the environments of interest with the same ℓ_2-norm bound κ used in the ENR formulation and C&W formulations. Therefore, high impact for the $\mathcal{A}_{\text{e}}^{\text{random}}$ and $\mathcal{A}_{\mathcal{M}}^{\text{random}}$ setting in these experiments indicates that we have identified a high-sensitivity direction. The results indicate that it is generally true that a direction of small margin corresponds to a high-sensitivity direction. However, the ENR formulation is more consistent in identifying high-sensitivity directions.

It is extremely surprising to notice that the impact of $\mathcal{A}^{\text{individual}}$ in JamesBond is distinctly lower than $\mathcal{A}_{\text{e}}^{\text{random}}$. The reason for this in JamesBond is that $\mathcal{A}_{\text{e}}^{\text{random}}$ consistently shifts all actions towards action 12 while $\mathcal{A}^{\text{individual}}$ causes the agent to choose different actions in every state. See section 5.2 for more details. We observe that this consistent shift towards one particular action results in a larger impact on the agent's performance in certain environments. In JamesBond, there are obstacles that the agent must jump over in order to avoid death, and consistently taking action 12 prevents the agent from jumping far enough. In CrazyClimber, the consistent shift towards one action results in the agent getting stuck in one state where choosing any other action would likely free it.[1]

We further investigated the state-of-the-art adversarially trained deep reinforcement learning policies as well as architectural differences in variants of DQN with our framework in Section 5.4 and in the full version of the paper respectively. We have found consistent results on identifying high-sensitivity directions independent from the training algorithms and the architectural differences.

5.2 Shifts in Actions Taken

In this subsection we investigate more closely exactly how perturbations along high-sensitivity directions affect the actions taken by the agent. We then argue that the results of this section suggest that high-sensitivity directions correspond to meaningful non-robust features used by the agent to make decisions. Note that in the reinforcement learning setting it is somewhat subtle to argue about the relationship between perturbations, non-robust features and actions taken. For example, suppose that the direction $\boldsymbol{v}$ does indeed correspond to a non-robust feature which the neural policy uses to decide which action to take in several states. The presence of the feature may indicate that different actions should be taken in state s versus state s', even though the feature itself is the same. Thus, even if $\boldsymbol{v}$ corresponds to a non-robust feature, this may not be detectable by looking at which actions the agent takes in state $s + \epsilon\boldsymbol{v}$ versus state s. However, the converse still holds: if there is a noticeable shift across multiple states towards one particular action a under the perturbation, then this constitutes evidence that the direction $\boldsymbol{v}$ corresponds to a non-robust feature that the neural policy uses to decide to take action a.

In the OpenAI Atari baseline version of the ALE, the discrete action set is numbered from 0 to $|\mathbb{A}|$. For each episode we collected statistics on the probability $P(a)$ of taking action a in the following scenarios:

- $P_{\text{base}}(a)$ - the fraction of states in which action $a \in \mathbb{A}$ is taken in an episode with no perturbation added.
- $P_{\text{shift}}(a)$ - the fraction of states in which action $a \in \mathbb{A}$ is taken in an episode with the perturbation added.

[1] See the full version of the paper for more detailed information on this issue, and the visualizations of the state representations of the deep reinforcement learning policies for these cases.

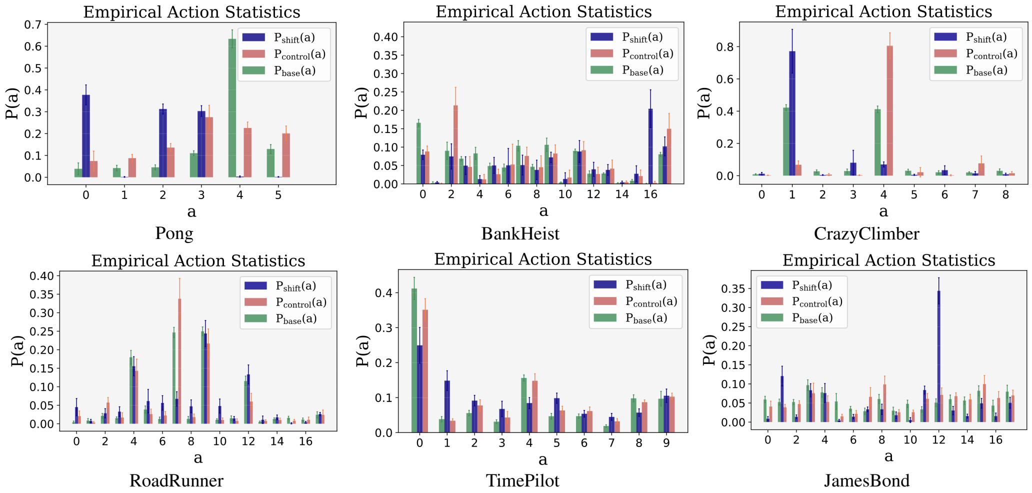

Figure 1: Action statistics for episode independent random state setting $\mathcal{A}_{\text{e}}^{\text{random}}$ and environment independent random state setting $\mathcal{A}_{\mathcal{M}}^{\text{random}}$ defined in Section 4 with ENR formulation for $P_{\text{control}}(a)$, $P_{\text{base}}(a)$ and $P_{\text{shift}}(a)$.

- $P_{\text{shift}}(a, b)$ - the fraction of states in which action $a \in \mathbb{A}$ *would have been* taken by the agent if there were no perturbation, but action $b \in \mathbb{A}$ was taken due to the added perturbation.
- $P_{\text{control}}(a) = \sum_b P_{\text{shift}}(a, b)$ - the fraction of states in which action $a \in \mathbb{A}$ *would have been* taken by the agent if there were no perturbation, in an episode with the perturbation added.

In Figure 1 we observe that in Roadrunner, JamesBond, and TimePilot $P_{\text{control}}(a)$ and $P_{\text{base}}(a)$ are similar. That is, for these environments the distribution on which action *would* be taken without perturbation does not change much. However, significant changes do occur in which action actually is taken due to the perturbation along the high-sensitivity direction. In Pong, BankHeist, and CrazyClimber the presence of the perturbation additionally causes a large change in which action *would* be taken.

In Figure 2 we show the heatmaps corresponding to $P_{\text{shift}}(a, b)$. In these heatmaps we show the fraction of states in which the agent would have taken action a, but instead took action b due to the perturbation. In some environments there is a dominant action shift towards one particular action b from one particular control action a as in BankHeist, CrazyClimber and TimePilot. The implication for these games is that the high-sensitivity direction $\boldsymbol{v}$ corresponds to a non-robust feature that consistently shifts the neural policy's decision from action a to action b.

In some games there are one or more clear bright columns where all the control actions are shifted towards the same particular action, for instance in JamesBond. The presence of a bright column indicates that, for many different input states (where the agent would have taken a variety of different actions), the high-sensitivity direction points across a decision boundary toward one particular action. As argued in the beginning of the section, these results suggest that the high-sensitivity direction found for JamesBond actually corresponds to a meaningful non-robust feature used by the neural policy. Closer examination of the semantics of RoadRunner reveals a similar shift towards a single type of action.

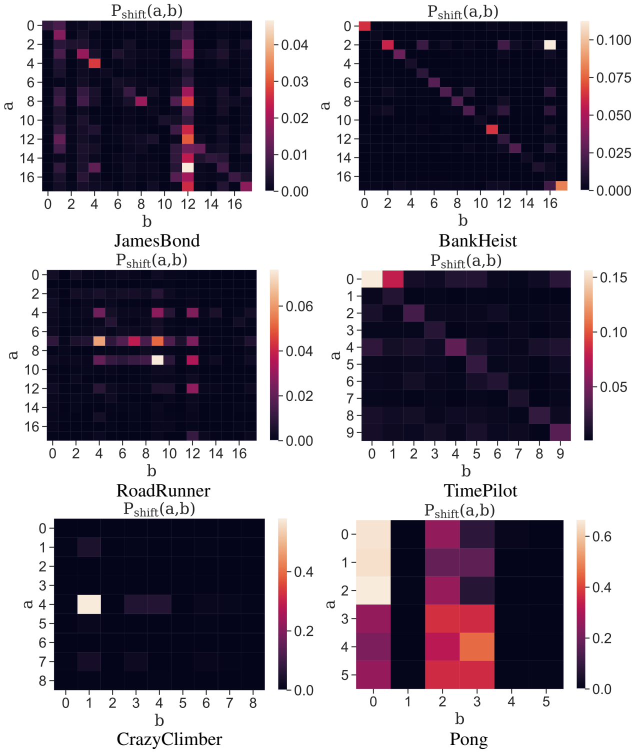

Figure 2: Heatmap of $P_{\text{shift}}(a, b)$ defined in Section 5.2 for episode independent random state setting $\mathcal{A}_{\text{e}}^{\text{random}}$ and environment independent random state setting $\mathcal{A}_{\mathcal{M}}^{\text{random}}$ with ENR.

In Roadrunner, the actions 4 and 12 correspond to left and leftfire, both of which move the player directly to the left[2]. These actions correspond to two bright columns in

[2] See the full version of the paper for more details on action

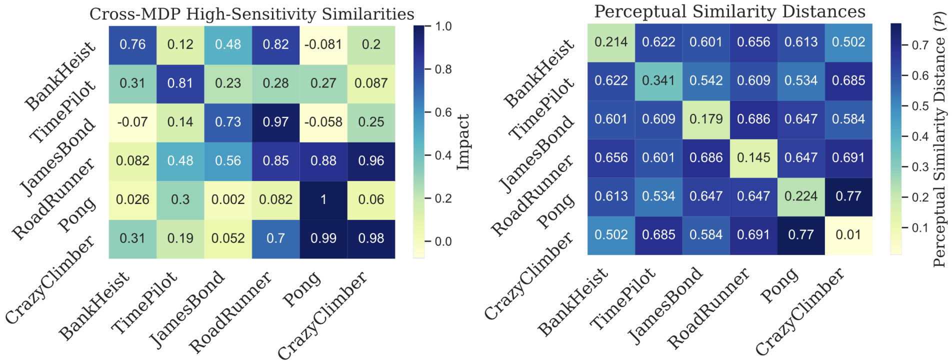

Figure 3: Cross-MDP high sensitivity similarities and perceptual similarities for $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ with ENR formulation.

the heatmap indicating significant shift towards the actions moving the player directly left from several other actions. As with the case of JamesBond, this suggests that the high-sensitivity direction $\boldsymbol{v}$ found in each of these games corresponds to a non-robust feature that the neural policy uses to decide on movement direction.

To make the results described above more quantitative we report the percentage shift towards a particular action type in Table 2. Specifically, we compute the total probability mass τ on actions which have been shifted from what they would have been, the total probability mass ρ shifted towards the particular action type, and report $\frac{\rho}{\tau}$.

MDP	Action Number	Percentage Shift $\left(\frac{\rho}{\tau}\right)$
BankHeist	Action 16	0.380
RoadRunner	Actions 4 and 12	0.279
JamesBond	Action 12	0.345
CrazyClimber	Action 1	0.708
TimePilot	Action 1	0.185

Table 2: Percentage of total action shift.

5.3 Cross-MDP Correlations in High-sensitivity Directions

In this section we investigate more closely the correlation of high-sensitivity directions between MDPs. In these experiments we utilize the environment independent random state setting $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ where the perturbation is computed from a random state of a random episode of one MDP, and then added to a completely different MDP. In Figure 3 we show the impacts for the $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ setting in six different environments. Each row shows the environment where the perturbation is added, and each column shows the environment from which the perturbation is computed. Note that the diagonal of the matrix corresponds to $\mathcal{A}^{\text{random}}_{e}$, and thus provides a baseline for the impact value of the high-sensitivity direction in the MDP from which it was computed. The off diagonal elements represent the degree to which the direction computed in one MDP remains a high-sensitivity direction in another MDP. Perceptual similarity distances are computed via LPIPS Zhang et al. (2018).

One can observe intriguing properties of the Atari baselines where certain environments are very likely to share high-sensitivity directions. For example, the high-sensitivity directions computed from RoadRunner are high-sensitivity directions in other MDPs and the high-sensitivity directions computed from other MDPs are high-sensitivity directions for RoadRunner. Additionally, in half of the environments higher or similar impact is achieved when using a direction from a different environment rather than using one computed in the environment itself. Recall that the results of Section 5.2 imply that high-sensitivity directions correspond to non-robust features within a single MDP. Therefore the high level of correlation in high-sensitivity directions across MDPs is an indication that deep reinforcement learning agents are learning representations that have correlated non-robust features across different environments.

5.4 Shared Non-Robust Features and Adversarial Training

In this section we investigate the correlations of the high-sensitivity directions between state-of-the-art adversarially trained deep reinforcement learning policies and vanilla trained deep reinforcement learning policies within the same MDP and across MDPs. In particular, Table 3 demonstrates the performance drop of the policies with settings $\mathcal{A}^{\text{Gaussian}}$, $\mathcal{A}^{\text{random}}_{\text{alg}}$ and $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ defined in Section 4. In more detail, Table 3 shows that a perturbation computed from a vanilla trained deep reinforcement learning policy trained in the CrazyClimber MDP decreases the performance by 54.6% when it is introduced to the observation system of the state-of-the-art adversarially trained deep reinforcement learning policy trained in the RoadRunner MDP. Similarly, a perturbation computed from a vanilla trained deep reinforcement learning policy trained in the CrazyClimber MDP decreases the performance by 65.9% when it is introduced to the observation system of the state-of-the-art adversarially trained deep reinforcement learning policy trained in the Pong MDP. This shows that non-robust features are not only shared across states and across MDPs, but also shared across different training algorithms. The state-of-the-

names and meanings for each game.

art adversarially trained deep reinforcement learning policies learn similar non-robust features which carry high sensitivity towards certain directions. It is quite concerning that algorithms specifically focused on solving adversarial vulnerability problems are still learning similar non-robust features as vanilla deep reinforcement learning training algorithms. This fact not only presents serious security concerns for adversarially trained models, but it posits a new research problem on the environments in which we train.

MDPs	$\mathcal{A}^{\text{Gaussian}}$	$\mathcal{A}^{\text{random}}_{\text{alg}}$	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$
RoadRunner	0.023±0.058	0.397±0.024	0.546±0.014
Pong	0.019±0.007	1.0±0.000	0.659±0.069
BankHeist	0.061±0.012	0.758±0.042	0.241±0.009

Table 3: Impacts of $\mathcal{A}^{\text{Gaussian}}$, $\mathcal{A}^{\text{random}}_{\text{alg}}$ and $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ where the perturbation is computed from a policy trained with DDQN and introduced to the observation system of the state-of-the-art adversarially trained deep reinforcement learning policy.

6 Conclusion

In this paper we focus on several questions: (i) Do neural policies share high sensitivity directions amongst different states? (ii) Are the high sensitivity directions correlated across MDPs and across algorithms? (iii) Do deep reinforcement learning agents learn non-robust features from the deep reinforcement learning training environments? To be able to investigate these questions we introduce a framework containing various settings. Using this framework we show that a direction of small margin to the decision boundary in a single state is often a high-sensitivity direction for the deep neural policy. We then investigate more closely how perturbations along high-sensitivity directions change the actions taken by the agent, and find that in some cases they shift the decisions of the policy towards one particular action. We argue that this suggests that high sensitivity directions correspond to non-robust features used by the policy to make decisions. Furthermore, we show that a high-sensitivity direction for one MDP is likely to be a high-sensitivity direction for another MDP in the Arcade Learning Environment. Moreover, we show that the state-of-the-art adversarially trained deep reinforcement learning policies share the exact same high-sensitivity directions with vanilla trained deep reinforcement learning policies. We systematically show that the non-robust features learnt by deep reinforcement learning policies are decoupled from states, MDPs, training algorithms and the architectural differences. Rather these high-sensitivity directions are a part of the learning environment resulting from learning non-robust features. We believe that this cross-MDP correlation of high-sensitivity directions is important for understanding the decision boundaries of neural policies. Furthermore, the fact that neural policies learn non-robust features that are shared across baseline deep reinforcement learning training environments is crucial for improving and investigating the robustness and generalization of deep reinforcement learning agents.

References

Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 274–283. PMLR.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, 253–279.

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

Carlini, N.; and Wagner, D. 2017. Towards Evaluating the robustness of neural networks. *In 2017 IEEE Symposium on Security and Privacy (SP)*, 39–57.

Chen, P.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C. 2018. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples. 10–17.

Dabney, W.; Ostrovski, G.; and Barreto, A. 2021. Temporally-Extended ϵ-Greedy Exploration. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dabney, W.; Rowland, M.; Bellemare, M. G.; and Munos, R. 2018. Distributional Reinforcement Learning With Quantile Regression. In McIlraith, S. A.; and Weinberger, K. Q., 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*, 2892–2901. AAAI Press.

Fedus, W.; Ramachandran, P.; Agarwal, R.; Bengio, Y.; Larochelle, H.; Rowland, M.; and Dabney, W. 2020. Revisiting Fundamentals of Experience Replay. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 3061–3071. PMLR.

Gleave, A.; Dennis, M.; Wild, C.; Neel, K.; Levine, S.; and Russell, S. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. *International Conference on Learning Representations ICLR*.

Goodfellow, I.; Shelens, J.; and Szegedy, C. 2015. Explaning and Harnessing Adversarial Examples. *International Conference on Learning Representations*.

Hasselt, H. v.; Guez, A.; and Silver, D. 2016. Deep Reinforcement Learning with Double Q-learning. *In Thirtieth AAAI conference on artificial intelligence*.

Hessel, M.; Modayil, J.; van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M. G.; and Silver, D. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. In McIlraith, S. A.;

and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 3215–3222. AAAI Press.

Huang, S.; Papernot, N.; Goodfellow, Y., Ian an Duan; and Abbeel, P. 2017. Adversarial Attacks on Neural Network Policies. *Workshop Track of the 5th International Conference on Learning Representations*.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples Are Not Bugs, They Are Features. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 125–136.

Kapturowski, S.; Ostrovski, G.; Quan, J.; Munos, R.; and Dabney, W. 2019. Recurrent Experience Replay in Distributed Reinforcement Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Korkmaz, E. 2020. 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.*

Korkmaz, E. 2021a. Adversarial Training Blocks Generalization in Neural Policies. *International Conference on Learning Representation (ICLR) Robust and Reliable Machine Learning in the Real World Workshop*.

Korkmaz, E. 2021b. Inaccuracy of State-Action Value Function for Non-Optimal Actions in Adversarially Trained Deep Neural Policies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2323–2327.

Korkmaz, E. 2021c. Investigating Vulnerabilities of Deep Neural Policies. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021*, volume 161 of *Proceedings of Machine Learning Research (PMLR)*, 1661–1670. AUAI Press.

Kos, J.; and Song, D. 2017. Delving Into Adversarial Attacks on Deep Policies. *International Conference on Learning Representations*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial examples in the physical world. *International Conference on Learning Representations (ICLR) Workshops*.

Levin, S.; and Carrie, J. 2018. Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian. *The Guardian*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. *International Conference on Learning Representations (ICLR)*.

Lin, Y.-C.; Zhang-Wei, H.; Liao, Y.-H.; Shih, M.-L.; Liu, i.-Y.; and Sun, M. 2017. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3756–3762.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous Methods for Deep Reinforcement Learning. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1928–1937. JMLR.org.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, a. G.; Graves, A.; Riedmiller, M.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533.

Pattanaik, A.; Tang, Z.; Liu, S.; and Gautham, B. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. *In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2040–2042.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust Adversarial Reinforcement Learning. *International Conference on Machine Learning ICML*.

Rowland, M.; Dadashi, R.; Kumar, S.; Munos, R.; Bellemare, M. G.; and Dabney, W. 2019. Statistics and Samples in Distributional Reinforcement Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 5528–5536. PMLR.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2016. Prioritized Experience Replay. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Schmitt, S.; Hessel, M.; and Simonyan, K. 2020. Off-Policy Actor-Critic with Shared Experience Replay. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 8545–8554. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347v2 [cs.LG]*.

Sun, J.; Zhang, T.; Xie, X.; Ma, L.; Zheng, Y.; Chen, K.; and Liu, Y. 2020. Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning. *Association for the Advancement of Artifical intelligence (AAAI)*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. *In Proceedings of the International Conference on Learning Representations (ICLR)*.

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On Adaptive Attacks to Adversarial Example Defenses. *NeurIPS*.

Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; and De Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. *Internation Conference on Machine Learning ICML.*, 1995–2003.

Xu, Z.; van Hasselt, H. P.; Hessel, M.; Oh, J.; Singh, S.; and Silver, D. 2020. Meta-Gradient Reinforcement Learning with an Objective Discovered Online. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D. S.; and Hsieh, C. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhang, R.; Isola, P.; Efros, A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Conference on Computer Vision and Pattern Recognition (CVPR)*.