DETECTING WORST-CASE CORRUPTIONS VIA LOSS LANDSCAPE CURVATURE IN DEEP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The non-robustness of neural network policies to adversarial examples poses a challenge for deep reinforcement learning. One natural approach to mitigate the impact of adversarial examples is to develop methods to detect when a given input is adversarial. In this work we introduce a novel approach for detecting adversarial examples that is computationally efficient, is agnostic to the method used to generate adversarial examples, and theoretically well-motivated. Our method is based on a measure of the local curvature of the neural network policy, which we show differs between adversarial and clean examples. We empirically demonstrate the effectiveness of our method in the Atari environment against a large set of state-of-the-art algorithms for generating adversarial examples. Furthermore, we exhibit the effectiveness of our detection algorithm with the presence of multiple strong detection-aware adversaries.

1 INTRODUCTION

Since Mnih et al. (2015) showed that deep neural networks can be used to parameterize reinforcement learning policies, there has been substantial growth in new algorithms and applications for deep reinforcement learning. While this progress has resulted in a variety of new capabilities for reinforcement learning agents, it has at the same time introduced new challenges due to the non-robustness of DNNs to imperceptible adversarial perturbations originally discovered by Szegedy et al. (2014). In particular, Huang et al. (2017); Kos & Song (2017) showed that the non-robustness of DNNs to adversarial perturbations extends to the deep reinforcement learning domain, where applications such as self-driving cars or automatic financial trading cannot tolerate such a vulnerability.

In the setting of image classification there has been significant effort to make DNNs robust to adversarial perturbations Goodfellow et al. (2015); Madry et al. (2018). At the same time, there is a line of work focused on showing the inevitability of adversarial examples and the intrinsic difficulty of learning robust classifiers Dohmatob (2019); Mahloujifar et al. (2019); Gourdeau et al. (2019). Given that it may not be possible to make DNNs completely robust to adversarial examples, a natural objective is to instead attempt to detect the presence of adversarial examples Pang et al. (2018); Yang et al. (2020); Cintas et al. (2020). Additional work has shown that many adversarial defense and detection methods fail in the white-box setting where the adversary is aware of both the trained model and the method used to detect examples Athalye et al. (2018); Carlini & Wagner (2017b).

In this paper we propose a novel detection method for adversarial examples in deep reinforcement learning. This is the first method for detecting adversarial examples in this setting, where computational efficiency is paramount since the method must be applied in real-time to every state encountered by the agent. Our approach relies on differences in the curvature of the neural policy in the neighborhood of an adversarial example when compared to a natural example. At a high level our method is based on the intuition that while natural examples have neighborhoods determined by an optimization procedure intended to learn a policy that works well across all states, each adversarial example is the output of some local optimization in the neighborhood of one particular state. Our proposed method is computationally efficient, requiring only one gradient computation and two policy evaluations, requires no training that depends on the adversarial attack method, and

is theoretically well-founded. In summary, we focus on detection of adversarial examples and make the following contributions:

- Our paper is the first to focus on detection of adversarial examples in the deep reinforcement learning domain.
- We propose a novel method, Detection of Adversaries with Taylor Approximation (DATA), to detect adversarial examples based on the local curvature of the neural network policy. DATA is computationally efficient, independent of the method used to generate the adversarial example, and theoretically justified.
- We conduct experiments in various games of the Arcade Learning Environment that demonstrate the effectiveness of DATA in detecting examples generated by several state-of-the-art adversarial attack methods.
- Furthermore, we demonstrate the effectiveness of DATA in the white-box setting where the adversary is aware of the detection method used.

2 Related Work and Background

Deep Reinforcement Learning: In this paper we focus on discrete action set Markov Decision Processes (MDPs) which are given by a continuous set of states $\mathbb{S}$, a discrete set of actions $\mathbb{A}$, a transition probability function $P : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$, and a reward function $r : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$. A policy $\pi : \mathbb{S} \to \mathcal{P}(\mathbb{A})$ assigns a probability distribution on actions $\pi(\cdot|s)$ to each state s. In deep reinforcement learning the policy π is parameterized by a deep neural network.

Adversarial Examples: Goodfellow et al. (2015) introduced the fast gradient method (FGM) for producing adversarial examples for image classification. The method is based on taking the gradient of the training cost function $J(x, y)$ with respect to the input image, and bounding the perturbation by ϵ where x is the input image and y is the output label. Later, an iterative version of FGM called I-FGM was proposed by Kurakin et al. (2016). This is also often referred to as Projected Gradient Descent (PGD) as in Madry et al. (2018) where the I-FGM update is

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

where $x_{\text{adv}}^{0} = x$. Dong et al. (2018) further modified I-FGM by introducing a momentum term in the update, yielding a method called MI-FGSM. Korkmaz (2020) later proposed a Nesterov-momentum based approach for the deep reinforcement learning domain. The DeepFool method of Moosavi-Dezfooli et al. (2016) is an alternative approach to those based on FGSM. DeepFool performs iterative projection to the closest separating hyperplane between classes. Another alternative approach proposed by Carlini & Wagner (2017a) is based on finding a minimal perturbation that achieves a different target class label. The approach is based on minimizing the loss

$$\min_{s^{\text{adv}} \in \mathbb{S}} c \cdot J(s^{\text{adv}}) + \left\| s^{\text{adv}} - s \right\|_2^2 \tag{2}$$

where s is the clean input, s_{adv} is the adversarial example, and $J(s)$ is a modified version of the cost function used to train the network. Chen et al. (2018) proposed a variant of the Carlini & Wagner (2017a) formulation that adds an ℓ_1-regularization term to produce sparser adversarial examples,

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

Adversarial Deep Reinforcement Learning: The adversarial problem initially has been investigated by Huang et al. (2017) and Kos & Song (2017) concurrently. In this work the authors show that perturbations computed via FGSM result in extreme performance loss on the learnt policy. Lin et al. (2017) and Sun et al. (2020) focused on timing strategies in the adversarial formulation and utilized the Carlini & Wagner (2017a) method to produce the perturbations. While there is a reasonable body of work focused on finding efficient and effective adversarial perturbations, a substantial body of work focused on building agents robust to these perturbations. Mandlekar et al. (2017) proposed to utilize FGSM perturbations during training time to obtain more robust agents. Pinto et al. (2017) modeled the adversarial interaction as a zero sum game and proposed a joint training strategy to increase robustness in continuous action space setting. Recently, Gleave et al. (2020) considered an

adversary who is allowed to take natural actions in a given environment instead of ℓ_p-norm bounded perturbations and modeled the adversarial relationship as a zero sum Markov game. However, recent concerns have been raised about adversarial training methods by Korkmaz (2021). In this paper the authors show that the state-of-the-art adversarial training techniques introduce a new set of non-robust features. Thus, with the rising concerns on robustness of recent proposed adversarial training techniques our work aims to solve the adversarial problem from a different perspective by detecting adversarial perturbations.

Detection of Adversarial Examples: There has been a long line of work on detection of adversarial examples for image classification. Metzen et al. (2017) proposed augmenting standard neural network image classifiers with a sub-network trained specifically to detect adversarial examples. However, this method has a cost of separately training a binary classifier to be able to detect the adversarial examples. Pang et al. (2018) proposed a modified neural network training strategy that is combined with a test-time thresholding method to distinguish adversarial from clean examples with additional training cost. Yang et al. (2020) studied detection of adversarial examples via feature attribution, utilizing the Leave-One-Out (LOO) attribution method proposed by Li et al. (2016). The LOO method is based on erasing each pixel of the image and observing how the output of the neural network changes, and so requires a number of neural network classifier evaluations equal to the resolution of the input image. Finally, Roth et al. (2019) suggest a statistical test based on measuring the average change in the odds ratio between classes under random perturbations. Evaluation of the neural classifier at hundreds of randomly perturbed samples are required in order for this method to get sufficiently accurate estimates of the average change in order to detect adversarial examples. In contrast to prior methods, our detection method does not require modifying the training of the neural network, does not require any training specific to the attack method used, and uses only two neural network function evaluations and one gradient computation.

3 Detection of Adversaries with Taylor Approximation (DATA)

In this section we give the high-level motivation for and formal description of our detection method. We begin by introducing necessary notation and definitions. We denote an original clean state by $\bar{s}$ and an adversarially perturbed state by s^{adv}.

Definition 3.1. *The* cost of a state*, $J(s, \tau)$, is defined as the cross entropy loss between the policy $\pi(a|s)$ of the agent, and a target distribution on actions $\tau(a)$.*

$$J(s, \tau) = -\sum_a \tau(a) \log(\pi(a|s)) \tag{4}$$

Definition 3.2. *The* argmax policy*, $\pi^*(a|s)$, is defined as the distribution which puts all probability mass on the highest weight action of $\pi(a|s)$.*

$$\pi^*(a|s) = \mathbb{1}(a = \arg\max_{a'} \pi(a'|s)) \tag{5}$$

We use the following notation for the gradient and Hessian with respect to states s:

$$\nabla_s J(s_0, \tau_0) = \nabla_s J(s, \tau)|_{s=s_0, \tau=\tau_0} \qquad \nabla_s^2 J(s_0, \tau_0) = \nabla_s^2 J(s, \tau)|_{s=s_0, \tau=\tau_0}$$

3.1 First-Order Detection of Adversaries with Taylor Approximation (FO-DATA)

As a naive baseline we first describe a detection method based on estimating how much the cost function $J(s, \tau)$ varies under small perturbations. Prior work on detection of adversarial examples Roth et al. (2019); Hu et al. (2019) has shown that the behavior of DNN classifiers under small, random perturbations is different at clean versus adversarial examples. Therefore, a natural baseline detection method is: given an input state s_0 sample a small random perturbation $\eta \sim \mathcal{N}(0, \epsilon I)$ and compute,

$$\mathcal{K}(s_0, \eta) = J(s_0 + \eta, \pi^*(\cdot|s_0)) - J(s_0, \pi^*(\cdot|s_0)). \tag{6}$$

The first-order detection method proceeds by first estimating the mean and the variance of $\mathcal{K}$ over a clean run of the agent in the environment. Next a threshold t is chosen so that a desired false positive rate (FPR) is achieved (i.e. some desired fraction of the states in the clean run lie more

than t standard deviations from the mean). Finally, at test time a state encountered by the agent is classified as adversarial if it is at least t standard deviations away from the mean. Otherwise the state is classified as clean. As a first attempt, the first-order method can be naturally interpreted as a finite-difference approximation to the magnitude of the gradient at s_0. If we assume that the first-order Taylor approximation of J is accurate in a ball of radius $r > \epsilon$ centered at s_0, then

$$J(s_0 + \eta, \pi^*(\cdot|s_0)) \approx J(s_0, \pi^*(\cdot|s_0)) + \nabla_s J(s_0, \pi^*(\cdot|s_0)) \cdot \eta.$$

Therefore,

$$\mathcal{K}(s_0, \eta) \approx \nabla_s J(s_0, \pi^*(\cdot|s_0)) \cdot \eta. \tag{7}$$

Thus, for $\eta \sim \mathcal{N}(0, \epsilon I)$ the test statistic $\mathcal{K}(s_0, \eta)$ is approximately distributed as a Gaussian with mean 0 and variance $\epsilon^2 \|\nabla_s J(s_0, \pi^*(\cdot|s_0))\|^2$. Under this interpretation one would expect the test statistics for clean and adversarial states to have the same mean with potentially different standard deviations, possibly making it hard to distinguish clean from adversarial. However, this is not what we observe empirically, and in fact the first-order method does a decent job of detecting adversarial examples. The method works because, in fact, the mean of $\mathcal{K}(\bar{s}, \eta)$ for clean examples $\bar{s}$ is reasonably well separated from the mean of $\mathcal{K}(s^{\text{adv}}, \eta)$ for adversarial examples s^{adv}. The empirical performance of the first-order method thus indicates that the assumption of accuracy for the first-order Taylor approximation of J does not hold in practice. This leads naturally to the consideration of information on the second derivatives of J in order to detect adversarial examples.

3.2 Second-Order Detection of Adversaries with Taylor Approximation (SO-DATA)

The second-order detection method is based on measuring the local curvature of the cost function $J(s, \tau)$. The method exploits the fact that $J(s, \tau)$ will have larger negative curvature at a clean example as compared to an adversarial example. In particular, the high level theoretical motivation for this approach is that adversarial examples are the output of a local optimization procedure which attempts to find a nearby perturbed state s^{adv} with a low value for the cost $J(s^{\text{adv}}, \tau)$ for some $\tau \neq \pi^*(\cdot|\bar{s})$. A direction of large negative curvature for $J(s^{\text{adv}}, \tau)$ indicates that a very small perturbation along this direction could dramatically decrease the cost function. Therefore, such points are likely to be unstable for local optimization procedures attempting to minimize the cost function in a small neighborhood. On the other hand, the curvature of $J(s, \tau)$ at a clean state $\bar{s}$ is determined by the overall algorithm used to train the deep reinforcement learning agent. This algorithm optimizes the parameters of the neural network policy while considering all states visited during training, and thus is not likely to be heavily overfit to the state $\bar{s}$. In particular, we expect larger negative curvature at $\bar{s}$ than at an adversarial example s^{adv}. We make the connection between negative curvature and instability for local optimization formal in Section 3.3. Based on the above discussion, a natural choice of metric for distinguishing adversarial versus clean examples is the most negative eigenvalue of the Hessian $\lambda_{\min}\left(\nabla_s^2 J(s_0, \pi^*(\cdot|s_0))\right)$. While this is the most natural measurement of curvature, it requires computing the eigenvalues of a matrix whose number of entries are quadratic in the input dimension. Since the input is very high-dimensional, and we would like to perform this computation in real-time for every state visited by the agent, computing the value $\lambda_{\min}$ is computationally prohibitive. Instead we approximate this value by measuring the curvature along a direction which is correlated with the negative eigenvectors of the Hessian. Given this direction, the value that we measure is the accuracy of the first order Taylor approximation of the cost of the given state $J(s, \tau)$. We denote the first order Taylor approximation at the state s_0 in direction η by

$$\tilde{J}(s_0, \eta) = J(s_0, \pi^*(\cdot|s_0)) + \nabla_s J(s_0, \pi^*(\cdot|s_0)) \cdot \eta.$$

The metric we will use to detect adversarial examples is the finite-difference approximation

$$\mathcal{L}(s_0, \eta) = J(s_0 + \eta, \pi^*(\cdot|s_0)) - \tilde{J}(s_0, \eta). \tag{8}$$

To see formally that Equation (8) gives an approximation of the most negative eigenvector of the Hessian, we will assume that the cost function $J(s, \tau)$ is well approximated by its second-order Taylor approximation at the point s_0 i.e.

$$J(s_0 + \eta, \pi^*(\cdot|s_0)) \approx J(s_0, \pi^*(\cdot|s_0)) + \nabla_s J(s_0, \pi^*(\cdot|s_0)) \cdot \eta + \eta^\top \nabla_s^2 J(s_0, \pi^*(\cdot|s_0))\eta \tag{9}$$

for a small enough perturbation η. Substituting the above formula into Equation (8) yields

$$\mathcal{L}(s_0, \eta) \approx \eta^\top \nabla_s^2 J(s_0, \pi^*(\cdot|s_0))\eta \tag{10}$$

Algorithm 1: SO-DATA

Input: Mean $\bar{\mathcal{L}}$ and variance $\sigma^2(\mathcal{L})$ from clean run. Detection threshold $t > 0$. Parameter $\epsilon > 0$.
for states s_i visited by agent **do**
  $\eta_i = \epsilon \dfrac{\text{sign}(\nabla_s J(s_i, \pi^*(\cdot|s_i)))}{\|\nabla_s J(s_i, \pi^*(\cdot|s_i))\|_2}$
  $\tilde{J}(s_i, \eta_i) = J(s_i, \pi^*(\cdot|s_i)) + \nabla_s J(s_i, \pi^*(\cdot|s_i)) \cdot \eta_i$
  $\mathcal{L}(s_i, \eta_i) = J(s_i + \eta_i, \pi^*(\cdot|s_i)) - \tilde{J}(s_i, \eta_i)$
  if $|\mathcal{L}(s_i, \eta_i) - \bar{\mathcal{L}}| > t \cdot \sigma(\mathcal{L})$ **then**
    Label state s_i as an adversarial example
  end if
end for

The above quadratic form is minimized when η lies in the same direction as the most negative eigenvector of the Hessian, in which case

$$\mathcal{L}(s_0, \eta) \approx \lambda_{\min}\left(\nabla_s^2 J(s_0, \pi^*(\cdot|s_0))\right) \|\eta\|_2^2 \tag{11}$$

We choose the sign of the gradient direction for measuring the accuracy of the first order Taylor approximation. To motivate this choice note that $-\nabla_s J(s, \tau)$ is locally the direction of steepest decrease for the cost function. If the gradient direction additionally has negative curvature of large magnitude, then small perturbations along this direction will result in even more rapid decrease in the cost function value than predicted by the first-order gradient approximation. Note that this can be true even if the gradient itself has small magnitude, as long as the negative curvature is large enough. Thus, by the discussion at the beginning of Section 3.2, adversarial examples are likely to have relatively smaller magnitude negative curvature in the gradient direction than clean examples. Formally, for $\epsilon > 0$ we set

$$\eta(s_0) = \epsilon \frac{\text{sign}\left(\nabla_s J(s_0, \pi^*(\cdot|s_0))\right)}{\|\nabla_s J(s_0, \pi^*(\cdot|s_0))\|_2}. \tag{12}$$

To calibrate the detection method we record the mean $\bar{\mathcal{L}} = \mathbb{E}_s[\mathcal{L}(s, \eta(s))]$ and variance $\sigma^2(\mathcal{L}) = \text{Var}_s[\mathcal{L}(s, \eta(s))]$ of our proposed test statistic over states from a clean run of the policy in the MDP. Then at test time we set a threshold $t > 0$, and for each state s_i visited by the agent test if

$$|\mathcal{L}(s_i, \eta(s_i)) - \bar{\mathcal{L}}| > t\sigma(\mathcal{L}). \tag{13}$$

If the threshold of t standard deviations is exceeded we classify the state s_i as adversarial, and otherwise classify it as clean. Pseudo-code for the second order method is given in Algorithm 1.

3.3 Negative Curvature and Instability of Local Optimization

In this section we formalize the connection between negative curvature and instability for local optimization procedures that motivated our definition of $\mathcal{L}(s, \eta)$. Given a state s_0 and a target distribution $\tau \neq \pi_*(\cdot|s_0)$, we assume the adversary is trying to find a state s^{adv} minimizing $J(s^{\text{adv}}, \tau)$ among all states close to s_0 by some metric. Formally, let $D_{s_0}(s) \geq 0$ be a convex function of s that should be thought of as measuring distance to s_0. One standard choice for the distance function is $D_{s_0}(s) = \|s - s_0\|_p^p$. We model the adversary as minimizing the loss

$$f(s) = J(s, \tau) + D_{s_0}(s). \tag{14}$$

In particular, we make the following assumption:

Assumption 3.1. *The adversarial state s^{adv} is a local minimum of $f(s)$.*

Of course this assumption is violated in practice since different adversarial attack methods optimize objective functions other than f, and do not necessarily always converge to a local minimum. Nevertheless the assumption allows us to make formal qualitative predictions about the behavior of the second-order detection method that correspond well with empirical results across a broad variety of adversarial attacks. We now state our main result lower bounding the curvature of $J(s^{\text{adv}}, \tau)$.

Proposition 3.1. *For $c > 0$ assume that the maximum eigenvalue of the Hessian $\nabla_s^2 D_{s_0}(s)$ is bounded by c. If s^* is a local minimum of $f(s)$ then $\lambda_{min}(\nabla_s^2 J(s^*, \tau)) \geq -c$*

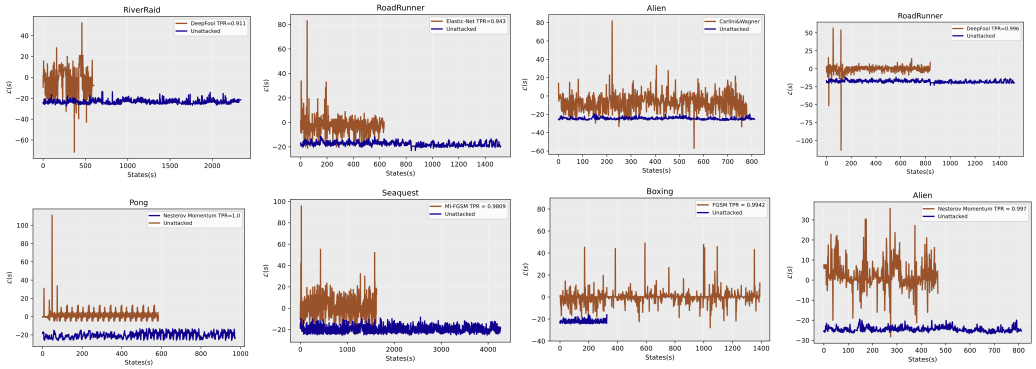

Figure 1: $\mathcal{L}(s)$ for our proposed method SO-DATA vs visited states with corresponding TPR values for the following attack methods: FGSM, MI-FGSM, Nesterov, DeepFool, Carlini&Wagner, Elastic Net Method. TPR values shown in the upper right box of the figure when FPR is equal to 0.01.

Proof. Let v be the eigenvector of $\nabla_s^2 J(s^*, \tau)$ corresponding to the minimum eigenvalue. At a local minimum s^* of $f(s)$ the Hessian $\nabla_s^2 f(s^*)$ must be positive semi-definite. Therefore,

$$\begin{aligned} 0 &\leq v^\top \nabla_s^2 f(s^*) v = v^\top \nabla_s^2 J(s^*, \tau) v + v^\top \nabla_s^2 D_{s_0}(s^*) v \\ &\leq \lambda_{\min}(\nabla_s^2 J(s^*, \tau)) + c \end{aligned}$$

Rearranging the above inequality completes the proof. □

In summary, the second order conditions for a local minimum of f imply a lower bound on the smallest eigenvalue of $\nabla_s^2 J(s^*, \tau)$. Thus, by Assumption 3.1, we obtain a lower bound on $\lambda_{\min}(\nabla_s^2 J(s^{\text{adv}}, \tau))$. The assumption that the maximum eigenvalue of the Hessian $\nabla_s^2 D_{s_0}(s)$ is bounded by c is satisfied for example when $D_{s_0}(s) = \frac{c}{2}\|s - s_0\|_2^2$. In contrast, the local curvature of the cost function $J(s, \tau)$ at a clean example is determined by an optimization procedure that updates the *weights* θ of the neural network policy rather than the states s. If we write $J_\theta(s, \tau)$ to make explicit the dependence on the weights, then the second order conditions for optimizing the original neural network apply to the Hessian with respect to weights $\nabla_\theta^2 J_\theta(s, \tau)$ rather than the Hessian with respect to states $\nabla_s^2 J_\theta(s, \tau)$. Additionally, first order optimality conditions can help to justify the choice of $\nabla_s J(s, \tau)$ as a good direction to check for negative curvature. Indeed by the first order conditions, at a local optimum s^* of $f(s)$ we have

$$0 = \nabla_s f(s^*) = \nabla_s J(s^*, \tau) + \nabla_s D_{s_0}(s^*). \tag{15}$$

Therefore, $\nabla_s J(s^*, \tau) = -\nabla_s D_{s_0}(s^*)$. So assuming the adversary finds a local optimum, $\nabla_s J(s, \tau)$ points in a direction that decreases the distance function $D_{s_0}(s^*)$. Thus sufficiently negative curvature in the direction of $\nabla_s J(s, \tau)$ implies not only that s is not a local minimum of f, but also that the distance function $D_{s_0}(s)$ can be decreased by moving along this direction of negative curvature. To summarize, we have shown that second order optimality conditions arising from computing an adversarial example give rise to lower bounds on the smallest eigenvalue of the Hessian $\lambda_{\min}\left(\nabla_s^2 J(s, \tau)\right)$. The function $\mathcal{L}(s, \eta)$ used to detect adversarial examples for SO-DATA is a finite difference approximation to

$$\eta^\top \nabla_s^2 J(s, \tau) \eta \geq \lambda_{\min}\left(\nabla_s^2 J(s, \tau)\right) \|\eta\|^2 .$$

Therefore the results of this section imply that $\mathcal{L}(s, \eta)$ should be larger at adversarial examples than clean examples.

4 Experiments

In our experiments agents are trained with DDQN Wang et al. (2016) in the Atari environment Bellemare et al. (2013) from OpenAI Brockman et al. (2016). For a baseline we compare FO-DATA and SO-DATA with the detection method of Roth et al. (2019), which is based on estimating the average change in the odds ratio between classes under noise. In Figure 1 we plot the value of

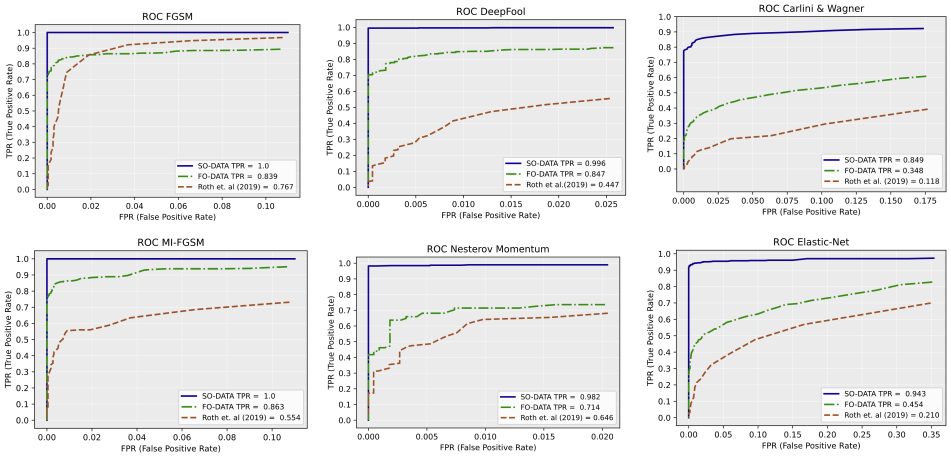

Figure 2: ROC curves of FO-DATA, SO-DATA and Roth et al. (2019) for the following attack methods: FGSM, MI-FGSM, Nesterov Momentum, DeepFool, Carlini&Wagner, Elastic Net Method in RoadRunner. TPR values shown in the lower right box of the figure when FPR is equal to 0.01.

Table 1: True Positive Rates (TPR) for FGSM, MI-FGSM, Nesterov Momentum, Carlini&Wagner, Elastic-Net and DeepFool when False Positive Rate (FPR) is equal to 0.01. The proposed methods SO-DATA and FO-DATA are evaluated, and compared with Roth et al. (2019) in Riverraid, RoadRunner, Alien, Seaquest, Boxing, Pong, and Robotank games. More results for different FPR values are reported in the supplementary material.

Detection Method-Attack Method	RiverRaid	RoadRunner	Alien	Seaquest	Boxing	Pong	Robotank
SO-DATA FGSM	0.997	1.0	1.0	0.995	0.994	1.0	0.999
FO-DATA FGSM	0.990	0.843	0.803	0.931	0.793	0.622	0.413
Roth et al. FGSM	0.681	0.767	0.885	0.403	0.264	0.424	0.911
SO-DATA M-IFGSM	0.998	1.0	1.0	0.985	0.910	1.0	0.985
FO-DATA M-IFGSM	0.952	0.863	0.991	0.981	0.827	0.622	0.470
Roth et al. M-IFGSM	0.775	0.554	0.929	0.581	0.499	0.679	0.777
SO-DATA Nesterov Momentum	0.995	0.989	0.996	0.952	0.865	1.0	0.954
FO-DATA Nesterov Momentum	0.990	0.714	0.997	0.979	0.746	0.633	0.574
Roth et al. Nesterov Momentum	0.785	0.646	0.925	0.671	0.517	0.687	0.753
SO-DATA Carlini&Wagner	0.910	0.988	0.945	0.723	0.856	0.850	0.713
FO-DATA Carlini&Wagner	0.695	0.594	0.642	0.516	0.785	0.494	0.119
Roth et al. Carlini&Wagner	0.036	0.118	0.018	0.004	0.016	0.028	0.032
SO-DATA Elastic Net	0.777	0.943	0.875	0.687	0.770	0.736	0.815
FO-DATA Elastic Net	0.685	0.454	0.561	0.502	0.743	0.361	0.212
Roth et al. Elastic Net	0.124	0.210	0.060	0.014	0.150	0.092	0.056
SO-DATA DeepFool	0.914	0.996	0.993	0.860	0.951	0.889	0.900
FO-DATA DeepFool	0.841	0.847	0.936	0.777	0.928	0.796	0.268
Roth et al. DeepFool	0.397	0.447	0.611	0.234	0.381	0.367	0.607

$\mathcal{L}(s)$ over states for various games without an adversarial attack and under adversarial attack with the following methods: Carlini & Wagner, Elastic Net, Nesterov Momentum, DeepFool, MIFGSM and FGSM. We show in the legends of Figure 1 the true positive rate (TPR) values for the different attacks when false positive rate (FPR) is equal to 0.01. The value of $\mathcal{L}(s)$ for clean states is generally well-concentrated and negative. On the other hand, for states computed by the different adversarial attack methods $\mathcal{L}(s)$ is clearly larger, matching the predictions of Proposition 3.1. The fact that $\mathcal{L}(s)$ is consistently larger at adversarial examples across a wide variety of adversarial perturbation methods indicates that Assumption 3.1 qualitatively captures the behavior of these methods. In particular the FGSM-based methods and DeepFool do not explicitly optimize an objective function of the form $f(s) = J(s, \tau) + D_{s_0}(s)$ as in Assumption 3.1. However, by enforcing a constraint on the distance of the adversarial example from the original clean example, these methods implicitly solve an optimization problem of the form given in (14), and thus exhibit the qualitative behavior predicted by Proposition 3.1.

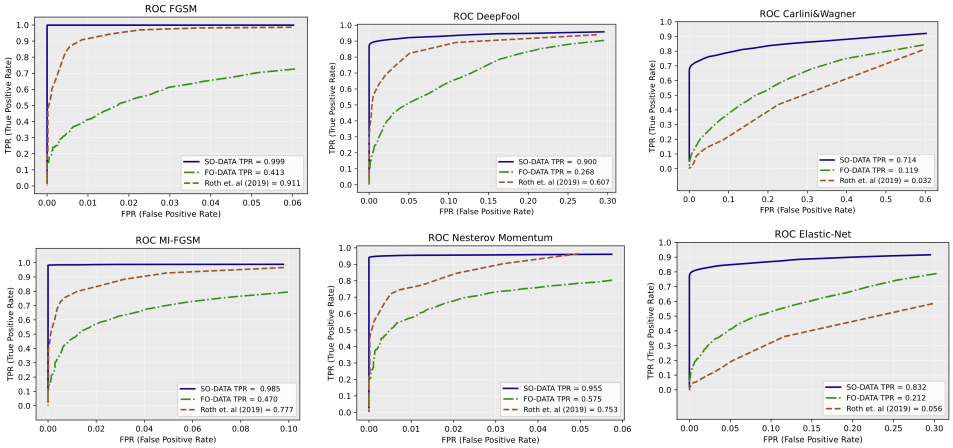

Figure 3: ROC curves of FO-DATA, SO-DATA and Roth et al. (2019) for the following attack methods: FGSM, MI-FGSM, Nesterov Momentum, DeepFool, Carlini&Wagner, Elastic Net Method in Robotank. TPR values are reported in the lower right box of the figure when FPR is equal to 0.01.

In Table 1 we show TPR values for FO-DATA, SO-DATA, and the Roth et al. (2019) method under the FGSM, MI-FGSM, Nesterov Momentum, DeepFool, Carlini&Wagner, and Elastic-Net attacks when FPR is equal to 0.01. For all of the attack methods in all of the environments SO-DATA is able to detect adversarial perturbations with large TPR. SO-DATA outperforms the other detection methods in all cases except for Nesterov Momentum in Alien and Seaquest where FO-DATA has TPR 0.997 and 0.980 while SO-DATA has 0.996 and 0.952. We also observe that while the perturbations computed by FGSM, MI-FGSM, Nesterov Momentum can generally be detected with large TPR values by all the detection methods, the perturbations computed by Carlini&Wagner and the Elastic-Net method are more difficult to detect. Despite the difficulty, SO-DATA achieves TPR values ranging from 0.713 to 0.988 for Carlini&Wagner, and TPR values ranging from 0.687 to 0.943 for Elastic-Net when FPR is equal to 0.01. In Figure 2 and Figure 3 we show ROC curves for each detection method under the FGSM, MI-FGSM, Nesterov Momentum, DeepFool, Carlini&Wagner and Elastic-Net method for RoadRunner and Robotank respectively. In Robotank the Roth et al. (2019) method outperforms FO-DATA and even approaches the TPR of SO-DATA for high FPR under FGSM, MI-FGSM, Nesterov Momentum and DeepFool. However for the Carlini&Wagner and Elastic-Net attacks, SO-DATA has a much higher TPR across a wide range of FPR levels.

5 Detection Aware Adversary

Recently, Tramer et al. (2020) introduced a comprehensive methodology for tailoring the optimization procedure used to produce adversarial examples in order to overcome detection and defense methods. In particular, the high level idea is to keep the attack as simple as possible while still accurately targeting the detection method. More specifically, the methodology is based on designing an attack based on gradient descent on some loss function. Further, minimizing the loss function should correspond closely to subverting the full detection method while still being possible to optimize. Critically, the authors highlight that while the choice of loss function to optimize can be a difficult task, the use of "feature matching" Gowal et al. (2019) can circumvent most of the current detection methods. We now describe how we applied the methodology from Tramer et al. (2020) to design detection aware adversaries for SO-DATA. As a first attempt, we tested the "feature matching" approach that was used to break the Roth et al. (2019) detection method in Tramer et al. (2020). This approach attempts to match the logits of the adversarial example to those of a clean example from a different class in order to evade detection. To optimize the loss for this method we used up to 1000 PGD iterations, and we searched step size varying from 0.01 to 10^{-6}. We find that this method succeeds in reducing the TPR of the Roth et al. (2019) method to nearly zero. It is also able to slightly reduce the TPR of our SO-DATA method (see results in Table 2). However, as we will see next, a larger reduction in the TPR of SO-DATA can be achieved by optimizing a modified version of the loss from Carlini & Wagner (2017b). Our next attempt is based on a modification of

Table 2: TPR for Feature Matching for SO-DATA and Roth et al. (2019) OAO FPR=0.01

Feature Matching	Riverraid	RoadRunner	Alien	Seaquest	Boxing	Robotank
SO-DATA	0.882	0.863	0.9016	0.955	0.988	0.8978
Roth et al. (2019)	0.0088	0.006	0.007	0.0146	0.0106	0.0158

the Carlini & Wagner (2017b) formulation to additionally minimize the cost function $\mathcal{L}(s)$ used in SO-DATA,

$$\min_{s^{\text{adv}} \in \mathbb{S}} c \cdot J(s^{\text{adv}}) + \left\| s^{\text{adv}} - s \right\|_2^2 + \lambda \cdot \mathcal{L}(s^{\text{adv}}). \tag{16}$$

Recall that $\mathcal{L}(s)$ is consistently larger at adversarial examples than at clean examples. Thus the above optimization problem attempts to find adversarial examples with as small values of $\mathcal{L}(s)$ as possible. Since the function $\mathcal{L}(s)$ involves taking the sign of the gradient, we use Backwards Pass Differentiable Approximation (BPDA) as introduced in Carlini & Wagner (2017b) to compute the gradients. However, we also tried designing an adversary with a fully differentiable cost function by using a perturbation in the gradient direction (without the sign). We found that this fully differentiable adversary performed significantly worse than the one based on BPDA. We conducted exhaustive grid search over all the parameters in this optimization method: learning rate, iteration number, confidence parameter κ, and objective function parameter λ. In C&W we used up to 30000 iterations to find adversarial examples to bypass detection methods. We searched the confidence parameter from 0 to 50, the learning rate from 0.001 to 0.1, and λ from 0.001 to 10. In our grid search over these hyperparameters we found that there is a trade-off between the attack success rate and the detection of the perturbations. In other words, if we optimize the perturbation to be undetectable the success rate of the perturbation (i.e. the rate at which the perturbation actually makes the agent choose a non-optimal action) decreases. Therefore, when finalizing the hyperparameters for the SO-DATA detection-aware adversary we restricted our search to a setting where the decrease in the success rate of the attack was at most 10%. Since FO-DATA is based on sampling a random

Table 3: TPR values of DATA in the presence of a detection aware adversary when FPR=0.01.

Detection Method	RiverRaid	RoadRunner	Alien	Seaquest	Boxing	Pong	Robotank
SO-DATA — C&W	0.650	0.849	0.445	0.381	0.710	0.712	0.657
FO-DATA — C&W	0.346	0.348	0.351	0.193	0.621	0.325	0.0973

perturbation, we use another approach introduced by Carlini & Wagner (2017b) to minimize the expectation of the original loss function when averaged over the randomness used in the detection method. In particular, we estimate the expectation by computing the empirical mean of the loss over 50 samples from the same noise source. As for the case of SO-DATA we grid search over hyperparameters to achieve as low a TPR as possible while losing at most 10% in the success rate of the attack. Table 3 shows the TPR in the adversary-aware setting with the best hyperparameters found for each method. The fact that SO-DATA still performs quite well in the adversary-aware setting is an indication that there is a fundamental trade-off between computing an adversarial example and minimizing $\mathcal{L}(s)$. This trade-off makes sense in light of Proposition 3.1, which shows that searching for an adversarial example in a small neighborhood will tend to increase $\mathcal{L}(s)$.

6 Conclusion

In this paper we introduced DATA, the first method for detection of adversarial attacks in deep reinforcement learning. Our method was theoretically motivated by the fact that local optimization objectives corresponding to the construction of adversarial examples lead naturally to lower bounds on the curvature of the cost function $J(s, \tau)$. We have further shown empirically that the curvature of $J(s, \tau)$ is significantly larger at adversarial examples than at clean examples, leading to a highly effective method SO-DATA for detecting adversarial examples in deep reinforcement learning. We additionally demonstrate that SO-DATA remains effective in the adversary-aware setting, and connect this fact to our original theoretical motivation. We believe that due to the strong empirical performance and solid theoretical motivation SO-DATA can be an important step towards producing robust deep reinforcement learning agents.

REFERENCES

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause (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*, pp. 274–283. PMLR, 2018.

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.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *In 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017a.

Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Bhavani M. Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha (eds.), *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pp. 3–14. ACM, 2017b.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 10–17. AAAI Press, 2018.

Celia Cintas, Skyler Speakman, Victor Akinwande, William Ogallo, Komminist Weldemariam, Srihari Sridharan, and Edward McFowland. Detecting adversarial attacks via subset scanning of autoencoder activations and reconstruction error. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 876–882. ijcai.org, 2020.

Elvis Dohmatob. Generalized no free lunch theorem for adversarial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1646–1654. PMLR, 09–15 Jun 2019.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

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.

Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (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*, pp. 7444–7453, 2019.

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy A. Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *https://arxiv.org/abs/1910.09338*, 2019.

Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q. Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (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*, pp. 1633–1644, 2019.

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.

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. Investigating vulnerabilities of deep neural policies. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

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.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL `http://arxiv.org/abs/1612.08220`.

Yen-Chen Lin, Hong Zhang-Wei, Yuan-Hong Liao, Meng-Li Shih, ing-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 3756–3762, 2017.

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.

Saeed Mahloujifar, Xiao Zhang, Mohammad Mahmoody, and David Evans. Empirically measuring concentration: Fundamental limits on intrinsic robustness. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (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*, pp. 5210–5221, 2019.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3932–3939, 2017.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2574–2582. IEEE Computer Society, 2016.

Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 4584–4594, 2018.

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

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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*, pp. 5498–5507. PMLR, 2019.

Jianwen Sun, Tianwei Zhang, Lei Xiaofei, Xie Ma, Yan Zheng, Kangjie Chen, and Yang. Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

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.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *NeurIPS*, 2020.

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.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. ML-LOO: detecting adversarial examples with feature attribution. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 6639–6647. AAAI Press, 2020.