Investigating Vulnerabilities of Deep Neural Policies

Ezgi Korkmaz[1]

[1]KTH Royal Institute of Technology , Stockholm, Sweden.

Abstract

Reinforcement learning policies based on deep neural networks are vulnerable to imperceptible adversarial perturbations to their inputs, in much the same way as neural network image classifiers. Recent work has proposed several methods to improve the robustness of deep reinforcement learning agents to adversarial perturbations based on training in the presence of these imperceptible perturbations (i.e. adversarial training). In this paper, we study the effects of adversarial training on the neural policy learned by the agent. In particular, we follow two distinct parallel approaches to investigate the outcomes of adversarial training on deep neural policies based on worst-case distributional shift and feature sensitivity. For the first approach, we compare the Fourier spectrum of minimal perturbations computed for both adversarially trained and vanilla trained neural policies. Via experiments in the OpenAI Atari environments we show that minimal perturbations computed for adversarially trained policies are more focused on lower frequencies in the Fourier domain, indicating a higher sensitivity of these policies to low frequency perturbations. For the second approach, we propose a novel method to measure the feature sensitivities of deep neural policies and we compare these feature sensitivity differences in state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies. We believe our results can be an initial step towards understanding the relationship between adversarial training and different notions of robustness for neural policies.

1 INTRODUCTION

Deep neural networks (DNNs) have led to notable progress across many areas of machine learning research and applications including computer vision Krizhevsky et al. [2012], natural language processing Sutskever et al. [2014], and speech recognition Hannun et al. [2014]. More recently deep neural networks (DNNs) have been employed in deep reinforcement learning by Mnih et al. [2015] to approximate the state-action value function for large action size or state size MDPs. With this initial success deep reinforcement learning became an emerging subfield with many applications such as robotics Kalashnikov et al. [2018], financial trading Noonan [2017] and medical Daochang and Jiang [2018], Huan-Hsin et al. [2017].

While the success of DNNs grew, a line of research focused on their reliability and robustness. Initially, Szegedy et al. [2014] demonstrated that it is possible to fool image classifiers by adding visually imperceptible perturbations to neural network inputs. Follow up work by Goodfellow et al. [2015] showed that these perturbations demonstrate that deep neural networks are learning linear functions. Several studies focused on overcoming this susceptibility towards specifically crafted visually imperceptible perturbations, and proposed training neural networks to be robust to these worst-case perturbations Madry et al. [2018]. However, adversarial training also has drawbacks: adversarially trained classifiers tend to have lower accuracy on standard inputs, and may be less robust to other types of distribution shift beyond worst-case ℓ_p-norm bounded perturbations Zhang et al. [2019]. While there is a significant amount of study focusing on adversarial training several works suggest that the existense of adversarial perturbations may be inevitable Dohmatob [2019], Mahloujifar et al. [2019], Gourdeau et al. [2019].

In this paper we focus on searching for answers to the following questions: (i) What are the sensitivity differences between state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies at a high

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1661–1670.

level? (ii) Does adversarial training move the sensitivities to worst-case ℓ_p-norm bounded distributional shift towards different directions in the input for deep neural policies? (iii) Does adversarial training create a new set of non-robust features while eliminating the existing ones? Therefore, in this paper we focus on examining the affects of adversarial training on neural policies in deep reinforcement learning and make the following contributions:

- We investigate the properties of adversarially trained neural policies via two different perspectives. The first is based on the response of adversarially trained policies to worst-case perturbations, and the second is based on probing adversarially trained policies via their sensitivity to features.
- For the worst-case perspective, we compare the frequency domain of the perturbations produced by Carlini and Wagner [2017] for adversarially trained models and vanilla trained models.
- We show that the perturbations produced from adversarially trained models are suppressed in high frequencies and more concentrated in lower frequencies in the Fourier domain compared to vanilla trained neural policies.
- For the sensitivity perspective, we propose a novel algorithm to detect feature based vulnerabilities of trained deep neural policies.
- We show that adversarially trained policies have a distinctive sensitivity pattern compared to vanilla trained deep neural policies. Furthermore, we demonstrate that while adversarially training removes the sensitivity of the neural policies towards some non-robust features, it also creates sensitivity towards a new set of non-robust features.

2 BACKGROUND

2.1 PRELIMINARIES

In this paper we focus on deep reinforcement learning for Markov decision processes (MDPs) given by a set of continuous states S, a set of discrete actions A, a transition probability distribution P on $S \times A \times S$, and a reward function $r : S \times A \to \mathbb{R}$. A policy $\pi : S \to \mathcal{P}(A)$ for an MDP assigns a probability distribution on actions to each $s \in S$. The goal for the reinforcement learning agent 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 $a_t \sim \pi(s_t)$. In Q-learning the learned policy is parametrized by a state-action value function $Q : S \times A \to \mathbb{R}$, which represents the value of taking action a in state s. Let $a^*(s) = \arg\max_a Q(s, a)$ denote the highest Q-value for an action in state s. The ϵ-greedy policy of the agent for Q-learning is given by taking action $a^*(s)$ with probability $1 - \epsilon$, and a uniformly random action with probability ϵ.

2.2 ADVERSARIAL EXAMPLES

Manipulating the output of neural networks by adding imperceptible perturbations was introduced by Szegedy et al. [2014] based on a box constrained optimization method. While this proposed method was computationally expensive, Goodfellow et al. [2015] proposed a faster and simpler method based on gradients in a nearby ϵ-ball,

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

where x represents the input, y represents the labels, and $J(x, y)$ represents the cost function used to train the network. Kurakin et al. [2016] further proposed an iterative search method inside this ϵ-ball using the fast gradient sign method (FGSM) proposed by Goodfellow et al. [2015].

$$x^0_{\text{adv}} = x, \tag{2}$$

$$x^{N+1}_{\text{adv}} = \text{clip}_\epsilon(x^N_{\text{adv}} + \alpha \text{sign}(\nabla_x J(x^N_{\text{adv}}, y))) \tag{3}$$

This method is also known as projected gradient descent (PGD) as in Madry et al. [2018]. Carlini and Wagner [2017] formulated the problem of producing adversarial perturbations in a more targeted way and proposed a method based on distance minimization for a given label in image classification. For deep reinforcement learning this formulation is based on distance minimization for a given a target action which is not equal to the best action decided by the trained policy,

$$\min_{s_{\text{adv}} \in D_{\epsilon,p}(s)} \|s_{\text{adv}} - s\|_p$$
$$\text{subject to} \quad \arg\max_a Q(s, a) \neq \arg\max_a Q(s_{\text{adv}}, a),$$

Note that $Q(s, a)$ denotes the state-action value function of the deep neural policy. Athalye et al. [2018] showed that the Carlini and Wagner [2017] adversarial formulation can break several proposed defenses. For this reason, in this paper we will focus on perturbations produced by the Carlini and Wagner [2017] formulation.

2.3 PERTURBATION FORMULATIONS AND ADVERSARIAL TRAINING

Initially adversarial examples were introduced in the deep reinforcement learning domain by Huang et al. [2017] and Kos and Song [2017] concurrently by utilizing FGSM as

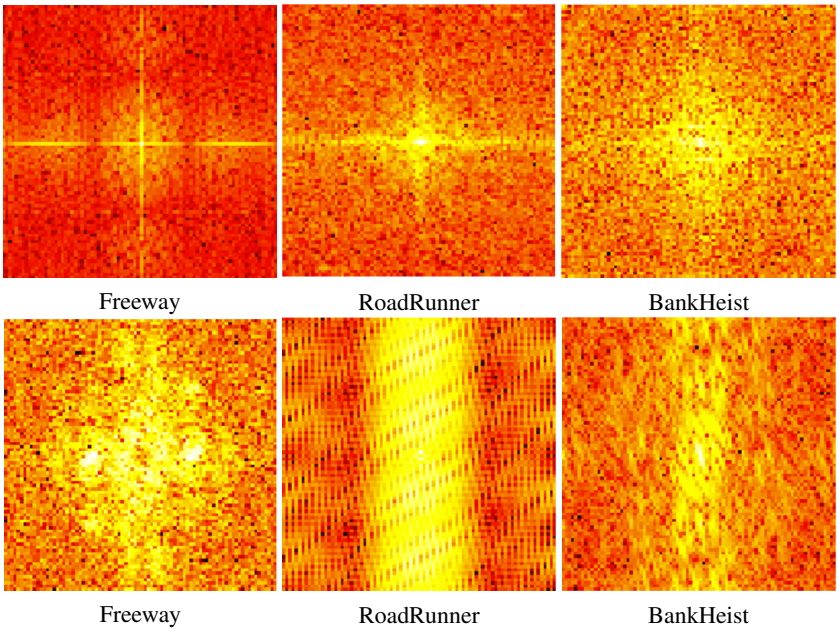


Figure 1: Fourier spectrum of the perturbations computed via Carlini and Wagner [2017] for state-of-the-art adversarially trained models and vanilla trained models. First Row: Adversarially trained. Second Row: Vanilla trained.

proposed by Goodfellow et al. [2015]. Several studies have been conducted to make deep reinforcement learning policies more robust to such specifically crafted malicious examples. Mandlekar et al. [2017] use adversarial examples produced by FGSM in the training data to regularize the policy in an attempt to increase robustness. Pinto et al. [2017] model the interaction between the perturbation maker and the agent as zero-sum Markov game and proposes a joint training algorithm to improve robustness against an adversary which aims to minimize the expected cumulative reward of the agent. Gleave et al. [2020] model the relationship between the adversary and the agent as zero-sum game where the adversary is limited to taking natural actions in the environment rather than minimal ℓ_p-norm bounded perturbations, and proposes an approach based on self-play to gain robustness against such an adversary. Quite recently Zhang et al. [2020] proposed a modified MDP called state-adversarial MDP with the aim of obtaining theoretically principled robust polices towards natural measurement errors and ℓ_p-norm worst case perturbations.

3 INVESTIGATING VULNERABILITIES

In this paper we aim to seek answers for several questions:

- What are the susceptibility differences between state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies?
- Do the sensitivities of deep neural policies shifts from worst-case ℓ_p-norm bounded perturbations towards different directions in the input with adversarial training?
- Does adversarial training create a new set of non-robust features while eliminating the existing ones?

In our experiments we use OpenAI Brockman et al. [2016] Atari baselines Bellemare et al. [2013]. Our models are trained with DDQN Wang et al. [2016] and SA-DDQN Zhang et al. [2020]. We test trained policies averaged over 10 episodes. Note that SA-DDQN is certified against ℓ_∞-norm bounded at $1/255$. Therefore, we also bound the perturbation by this threshold and find the perturbations with ℓ_∞-norm lower than this value.

4 NEURAL POLICY PERTURBATIONS IN THE FOURIER DOMAIN

For the adversarially trained agents, we focus on the state-of-the-art adversarial training algorithm SA-DQN proposed by Zhang et al. [2020]. In this study the authors model the interaction between the neural policy and the introduced perturbations as a state-adversarial modified Markov Decision Process (MDP). The authors claim that the agents trained in SA-MDP with the proposed algorithm SA-DQN are more robust to adversarial perturbations and natural noise introduced to the agent's perception system. Furthermore, the authors demonstrate the robustness of SA-DQN against perturbations produced by the PGD attack proposed by Madry et al. [2018].

In this section we conduct an investigation on the frequency

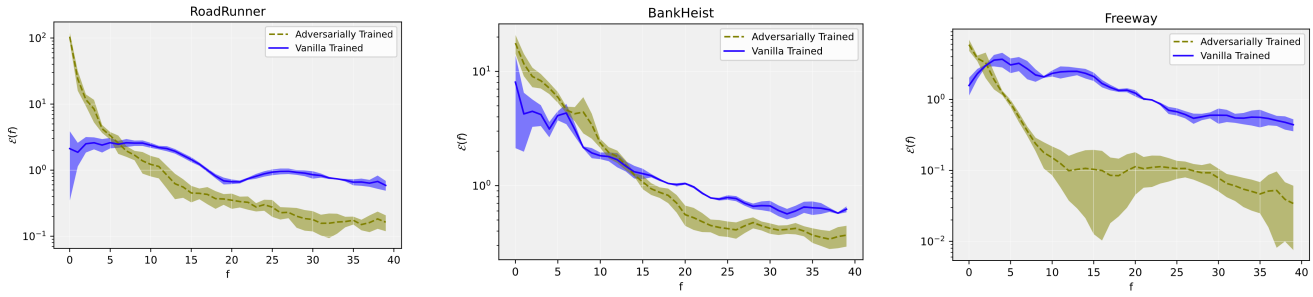


Figure 2: Power spectrum of the perturbations computed via Carlini and Wagner [2017] for adversarially trained models and vanilla trained models in Fourier domain for RoadRunner, BankHeist and Freeway.

domain of the perturbations computed from vanilla trained agents and adversarially trained agents. In particular, we compute a minimum length perturbation via Carlini and Wagner [2017] which causes the agent to change its optimal learned action. We found that the certified defence proposed by Zhang et al. [2020] can be overcome via Carlini and Wagner [2017] for the certified bound given in Zhang et al. [2020]. For these minimal perturbations we compute the Fourier transform of the perturbation and record this data. By comparing the results of this experiment for adversarially trained versus vanilla trained agents, we can understand the affects of adversarial training on the directions to which the neural policy is sensitive. We now describe the details of the experimental setup.

In more detail, we ensure that the perturbations $\eta = s_{\text{adv}} - s$ produced by Carlini and Wagner [2017] satisfies two requirements:

- The optimal action in state s changes i.e. $a^*(s) \neq a^*(s_{\text{adv}})$.
- The perturbation is bounded by the certified defense level proposed by Zhang et al. [2020] $\|\eta\|_\infty < \frac{1}{255}$.

In Figure 1 we visualize the Fourier transform of a minimal perturbation for both vanilla trained and adversarially trained agents in RoadRunner, BankHeist, and Freeway. The center of each image corresponds to the Fourier basis function where both spatial frequencies are zero, and the magnitude of the spatial frequencies increases as one moves outward from the center. There is a distinctive difference between vanilla and adversarially trained agents in the qualitative appearance of Fourier transforms of the minimal pertubations. Further, from these visualizations it is clear that the perturbations for the adversarially trained models generally have their Fourier transform concentrated at lower frequencies than those of the vanilla trained models.

To make this claim formal, for each number f we compute the total energy $\mathcal{E}(f)$ of the Fourier transform for basis functions whose maximum spatial frequency is equal to f. In Figure 2 we plot the average of $\mathcal{E}(f)$ over the minimal perturbations computed in our experiments. We find that the minimal perturbations computed for adversarially trained neural policies are indeed shifted towards lower frequencies when compared to those for vanilla trained neural policies. This shift in the frequency domain of the computed perturbations implies that adversarially trained neural policies may be more sensitive towards low frequency perturbations.

5 VISUALIZING NEURAL POLICY VULNERABILITIES

In this section we propose two different methods to visualize vulnerabilities of deep neural policies to their input observations. First, we describe our proposed method of feature vulnerability mapping KMAP in detail. To be able to visualize weaknesses we record the drop in the state-action value $Q(s, a)$ caused by setting each pixel in s to zero one at a time. In particular, let $Z_{i,j} : S \to S$ be the function which sets the i, j coordinate of s to zero and leaves the other coordinates unchanged. We define,

$$\mathcal{K}(i, j) = Q(s, a^*) - Q(s, \arg\max_a Q(Z_{i,j}(s), a)). \quad (4)$$

Note that the difference in Equation 4 represents the drop in the Q-value in state s, when taking the optimal action for the state $Z_{i,j}(s)$. Therefore, $\mathcal{K}(i, j)$ aims to measure the drop in the Q-values of the neural policy with respect to individual pixel changes. In other words, $\mathcal{K}(i, j)$ is a mapping of features to an importance metric determined by the deep neural policy. We describe our proposed KMAP method in detail in Algorithm 1.

As a natural point of comparison we propose another algorithm HMAP to visualize input based vulnerabilities. In particular, HMAP is based on measuring the effect of each individual pixel on the decision of the deep neural policy by measuring the cross-entropy loss between $\pi(s, a)$ and $\pi(Z_{i,j}(s), a)$.

Algorithm 1: KMAP Feature vulnerability mapping

Input: State-action value function $Q(s,a)$,actions a, states s, T_d size of the dimension d of the state s, and $s(i,j)$ is the value of the i,j-th pixel in state s.
Output: Visual weakness mapping function $\mathcal{K}(i,j)$
$s_{\text{aug}} = s$
for $i = 1$ **to** T_1 **do**
  for $j = 1$ **to** T_2 **do**
    $s_{\text{aug}}(i,j) = 0$
    a^*_{aug} = argmax$_a$ $Q(s_{\text{aug}}, a)$
    a^* = argmax$_a$ $Q(s,a)$
    $\mathcal{K}(i,j)$ += $Q(s,a) - Q(s, a^*_{\text{aug}})$
    $s_{\text{aug}} = s$
  end for
end for
Return: $\mathcal{K}(i,j)$

$$\mathcal{H}(i,j) = -\sum_{a \in A} \pi(s,a) \log(\pi(Z_{i,j}(s), a)) \tag{5}$$

where we compute the policy $\pi(s,a)$ via the softmax of the state-action value function $Q(s,a)$,

$$\pi(s,a) = \frac{e^{Q(s,a)/T}}{\sum_{a \in A} e^{Q(s,a)/T}}. \tag{6}$$

Note that T represents the temperature constant. We describe the HMAP method in detail in Algorithm 2.

6 RESULTS ON KMAP AND HMAP

In Figure 3 we show the KMAP and HMAP heatmaps for a state-of-the-art adversarially trained neural policy and vanilla trained neural policy in RoadRunner. One intriguing observation from the KMAP heatmap for the adversarially trained deep neural policy is the vulnerability to pixel changes in a certain column shown in Figure 3. In comparison, the vanilla trained agent's vulnerability is concentrated on several rows in a different part of the input. Another interesting fact about Figure 3 is that the vulnerability pattern for the vanilla trained agent is concentrated on a portion of the input image with which the agent does not interact during the game. In fact, in RoadRunner, the vulnerability pattern for the vanilla trained agent is in a portion of the input that the agent is not able to even visit.

Figure 5 the BankHeist KMAP $\mathcal{K}(i,j)$ results show a similar sensitivity pattern between the adversarially trained deep neural policy and the vanilla trained deep neural policy. In particular adversarially trained KMAP $\mathcal{K}(6:10, 29:31)$[1] is quite similar to vanilla trained $\mathcal{K}(3:7, 23:25)$[2]. Thus,

Algorithm 2: HMAP Feature vulnerability mapping

Input: State-action value function $Q(s,a)$, actions a, states s, policy $\pi(s,a)$, T_d size of the dimension d of the state s, and $s(i,j)$ is the value of the i,j-th pixel in state s.
Output: Visual weakness mapping function $\mathcal{H}(i,j)$
$s_{\text{aug}} = s$
for $i = 1$ **to** T_1 **do**
  for $j = 1$ **to** T_2 **do**
    $s_{\text{aug}}(i,j) = 0$
    $\pi(s,a) = \text{softmax}(Q(s,a))$
    $\pi(s_{\text{aug}}, a) = \text{softmax}(Q(s_{\text{aug}}, a))$
    $\mathcal{H}(i,j)$ +=$-\sum_{a \in A} \pi(s,a) \log(\pi(s_{\text{aug}}, a))$
    $s_{\text{aug}} = s$
  end for
end for
Return: $\mathcal{H}(i,j)$

in this setting we observe that the vulnerabilities towards a certain set of features remians the same with adversarial training.

Figure 8 and Figure 7 show heatmaps of KMAP $\mathcal{K}(i,j)$ and HMAP $\mathcal{H}(i,j)$ for Freeway. We observe that while the KMAP $\mathcal{K}(i,j)$ pattern for the vanilla trained agent lies on the portion of the input where the optimal policy is executed by the agent, the KMAP $\mathcal{K}(i,j)$ for the adversarially trained deep neural policy has a straightforward grid pattern. Based on these results, we hypothesize that adversarial training decouples vulnerability from the features relevant to the optimal policy learned by the agent.

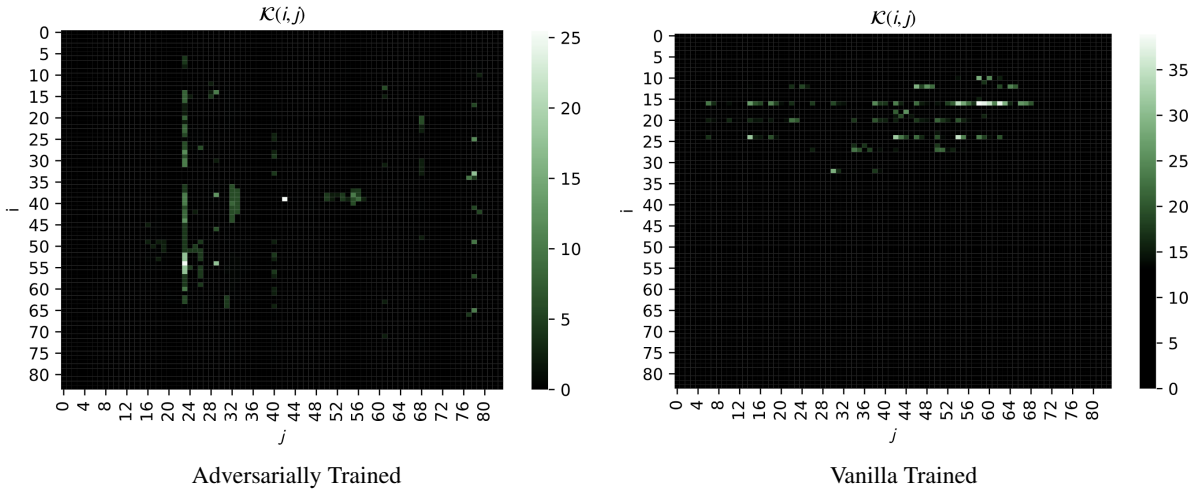
The decoupling of relevant features and vulnerability can be seen as an additional way in which adversarial training shifts the vulnerabilities of deep neural policies. This complements the results of Section 5, where we observe a vulnerability shift by looking at worst-case ℓ_p-norm bounded perturbations, and observing that these perturbations are more concentrated on lower frequencies in adversarially trained agents.

While visual observation indicates very different vulnerability patterns for these two disjoint training strategies, we also introduce a quantitative metric to compare the results of KMAP and HMAP for vanilla and adversarially trained agents. In particular, we use the ratio of the ℓ_1 and ℓ_2 norms to measure the sparsitiy via,

$$S(\mathcal{K}) = \frac{\|\mathcal{K}\|_1}{\|\mathcal{K}\|_2}. \tag{7}$$

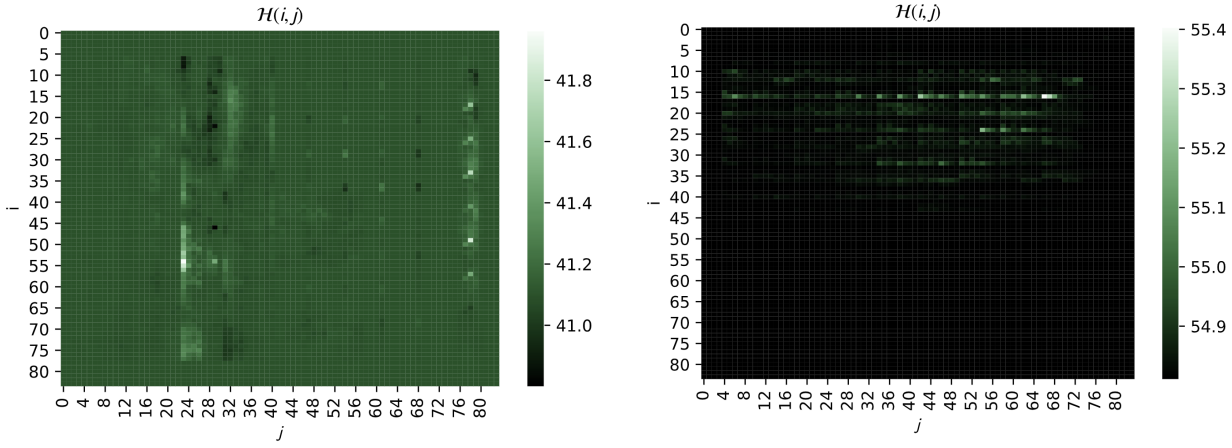
[1] $\mathcal{K}(k:m, l:n)$ denotes $\mathcal{K}(i,j)$ values for the set of coordinates $i \in \{k, \ldots, m\}, j \in \{l, \ldots, n\}$.

[2] This portion of the input observation corresponds to the fuel gauge in BankHeist. In this game the player loses a life when the fuel runs out.



Adversarially Trained Vanilla Trained

Figure 3: KMAP $\mathcal{K}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in RoadRunner. Left: Adversarially trained. Right: Vanilla trained.



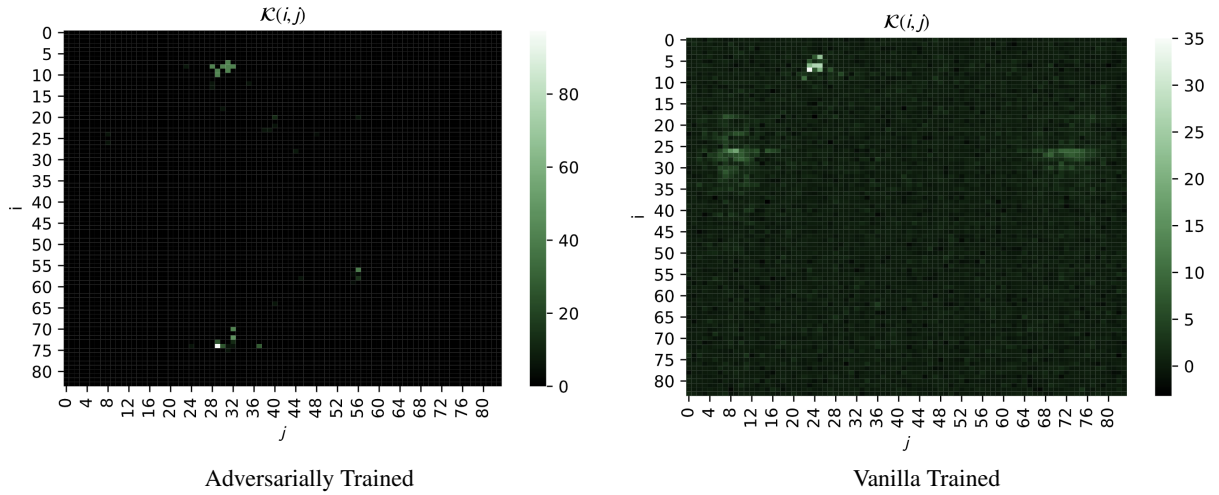
Adversarially Trained Vanilla Trained

Figure 4: HMAP $\mathcal{H}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in RoadRunner. Left: Adversarially trained. Right: Vanilla trained.

Here smaller values of $S(\mathcal{K})$ correspond to sparser vulnerability patterns. We also measure how spread out the vulnerability pattern is via the entropy of the softmax of $\mathcal{K}(i,j)$,

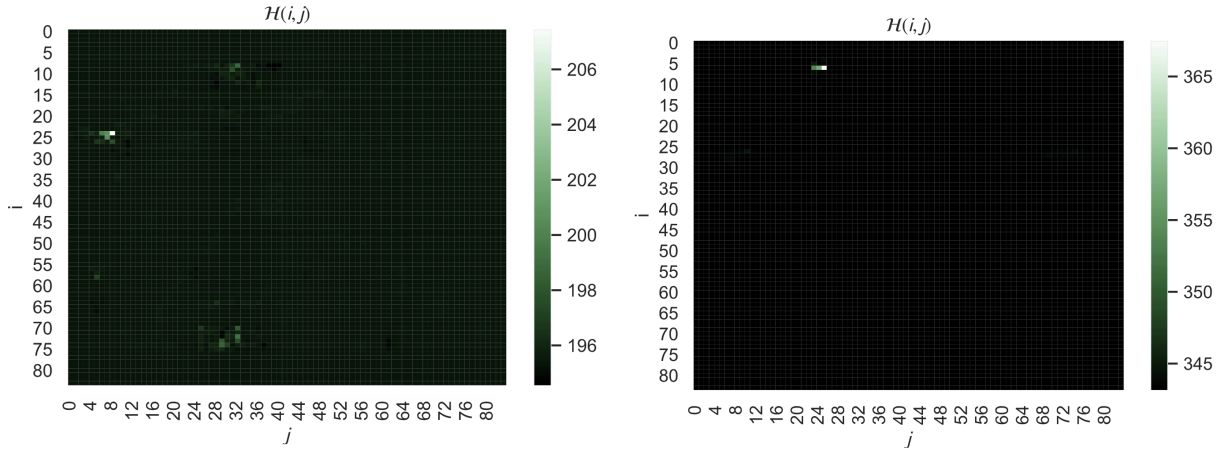
$$H(\mathcal{K}) = -\sum_{i,j} \text{softmax}(\mathcal{K})_{i,j} \log(\text{softmax}(\mathcal{K})_{i,j}) \quad (8)$$

In Table 1 and Table 2 we show the sparsity and entropy results respectively for KMAP $\mathcal{K}(i,j)$ and HMAP $\mathcal{H}(i,j)$ for adversarially trained deep neural policies and vanilla trained deep neural policies. We observe that for KMAP the vulnerability of adversarially trained models with respect to features are more sparse than the vanilla trained agents. The results for HMAP are more mixed, and it is often barely possible to detect the sparsity difference via $S(\mathcal{H})$ and only possible in half of the games via $H(\mathcal{H})$. In general, KMAP $\mathcal{K}(i,j)$ provides a better estimation of sensitivity of deep neural policies to individual pixel changes than HMAP $\mathcal{H}(i,j)$. While KMAP $\mathcal{K}(i,j)$ captures the actual impact of the feature change on the decision of the deep neural policy HMAP $\mathcal{H}(i,j)$ captures the difference between the softmax policy distributions $\pi(s,a)$ and $\pi(Z_{i,j}(s),a)$, which do not necessarily correspond to the decisions made by the neural policy.



Adversarially Trained Vanilla Trained

Figure 5: KMAP $\mathcal{K}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in BankHeist. Left: Adversarially trained. Right: Vanilla trained.



Adversarially Trained Vanilla Trained

Figure 6: HMAP $\mathcal{H}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in BankHeist. Left: Adversarially trained. Right: Vanilla trained.

Table 1: Sparsity results of KMAP $\mathcal{K}(i,j)$ and HMAP $\mathcal{H}(i,j)$ for adversarially trained and vanilla trained deep neural policies.

Training Method Sparsity	Vanilla Trained $S(\mathcal{K})$	Adversarially Trained $S(\mathcal{K})$	Vanilla Trained $S(\mathcal{H})$	Adversarially Trained $S(\mathcal{H})$
Freeway	53.7272	20.4641	83.9999	83.91587
BankHeist	38.4085	4.8812	83.99999	83.99994
RoadRunner	33.1493	11.3216	83.999992	83.999993
Pong	49.7993	1.9508	83.99999	83.9999

7 CONCLUSION

In this paper we focused on investigating the vulnerabilities of deep neural policies with respect to their inputs. We examined the vulnerability shifts between state-of-the-art adversarially trained deep neural policies and vanilla trained policies. First, we investigate through worst-case ℓ_p-norm bounded distributional shift. We explored and com-

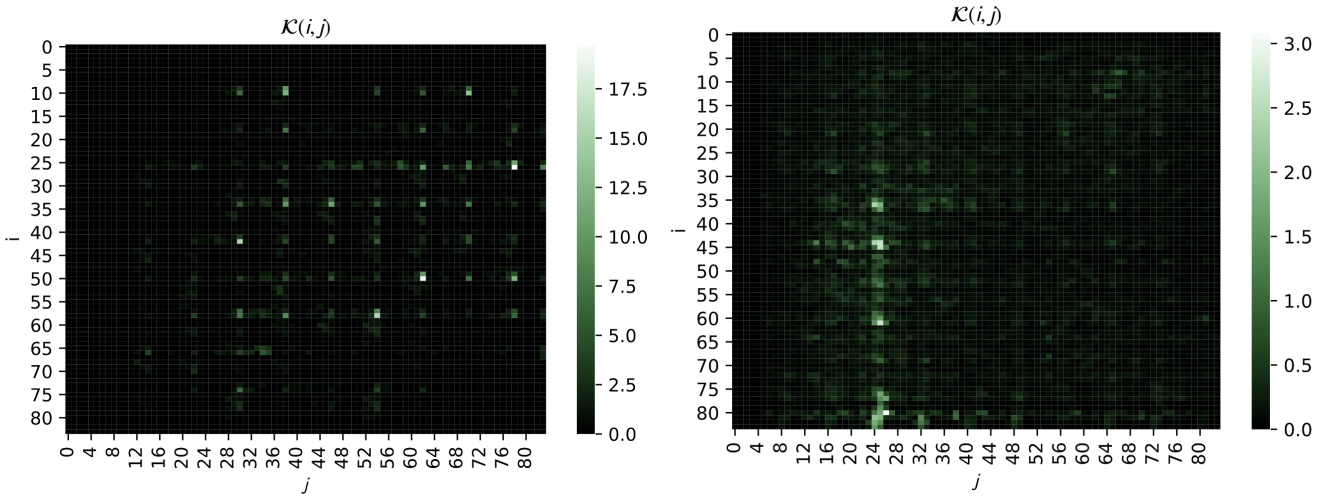


Figure 7: KMAP $\mathcal{K}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in Freeway. Left: Adversarially trained (SA-DDQN). Right: Vanilla trained (DDQN).

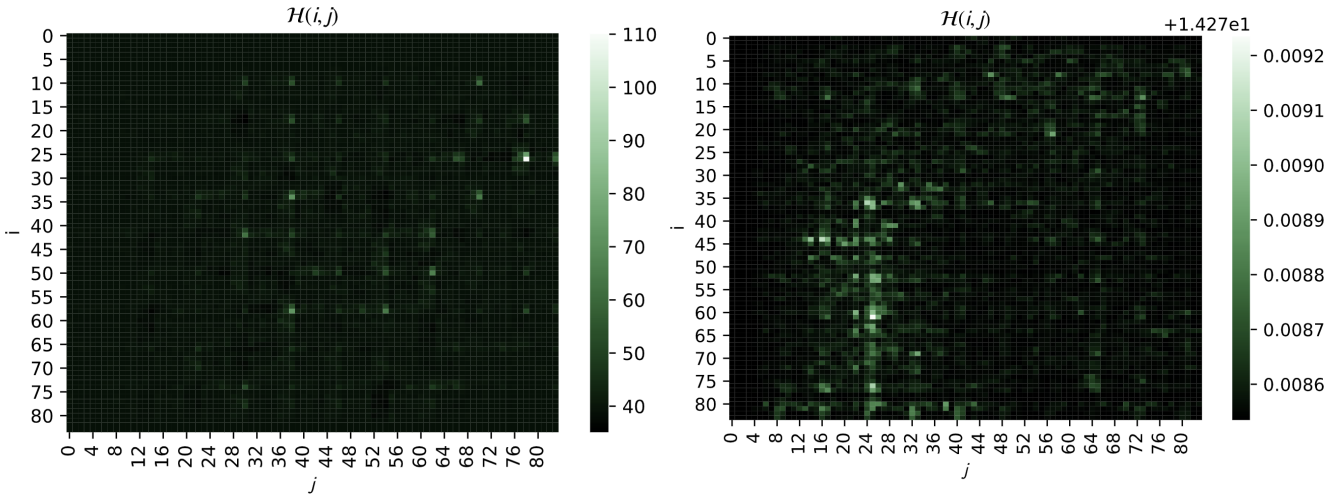


Figure 8: HMAP $\mathcal{H}(i,j)$ heatmaps for state-of-the-art adversarially trained deep neural policy and vanilla trained deep neural policy in Freeway. Left: Adversarially trained (SA-DDQN). Right: Vanilla trained (DDQN).

Table 2: Entropy of KMAP $\mathcal{K}(i,j)$ and HMAP $\mathcal{H}(i,j)$ for adversarially trained deep neural policies and vanilla trained deep neural policies.

Training Method Entropy	Vanilla Trained $H(\mathcal{K})$	Adversarially Trained $H(\mathcal{K})$	Vanilla Trained $H(\mathcal{H})$	Adversarially Trained $H(\mathcal{H})$
Freeway	8.8287	0.0807	8.8616	0.5677
BankHeist	0.0055	$1.542e^{-20}$	8.8616	0.54405
RoadRunner	1.2346	0.6973	8.8610	8.8608
Pong	8.8615	8.5658	8.8616	8.8615

pared the frequency domain of the perturbations computed from state-of-the-art adversarially trained neural policies and vanilla trained neural policies. We found that the perturbations computed from adversarially trained models were more concentrated in lower frequencies compared to the vanilla trained neural policies. Second, we propose two dif-

ferent algorithms that we call KMAP and HMAP to detect vulnerabilities with respect to input in deep neural policies. We compare the state-of-the-art adversarially trained neural policies and vanilla trained neural policies with our proposed methods KMAP and HMAP via several experiments in various environments. We found that while adversarial training removes sensitivity to certain features, it builds sensitivity towards a new set of features. We believe this work lays out the vulnerabilities of adversarially trained neural policies in a systematic way, and can be an initial step towards building robust and reliable 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, editors, *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*, pages 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.*, page 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)*, pages 39–57, 2017.

Liu Daochang and Tingting. Jiang. Deep reinforcement learning for surgical gesture segmentation and classification. *In International conference on medical image computing and computer-assisted intervention.*, pages 247–255.Springer, Cham, 2018.

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

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, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7444–7453, 2019.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Diamos Greg, Erich Else, Ryan Prenger, Sanjeev Satheesh, Sengupta Shubho, Ada Coates, and Andrew Ng. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

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.

Tseng Huan-Hsin, Sunan Cui, Yi Luo, Jen-Tzung Chien, Randall K. Ten Haken, and Issam El. Naqa. Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical physics 44*, pages 6690–6705, 2017.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012.

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

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, editors, *Advances in Neural Information Processing Systems*

32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 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)*, pages 3932–3939, 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.

Laura Noonan. Jpmorgan develops robot to execute trades. *Financial Times*, page 1928–1937, July 2017.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. . Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 2014.

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.

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.*, page 1995–2003, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *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*, pages 7472–7482. PMLR, 2019.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane S. Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.