Adversarial Attacks on Neural Network Policies

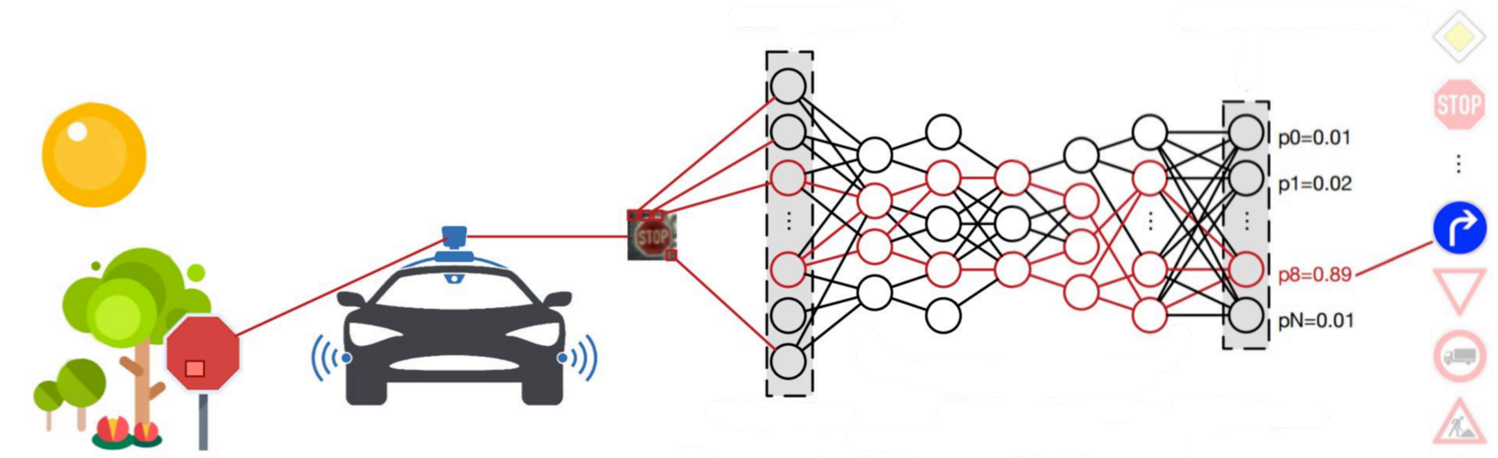
Sandy Huang[1], Nicolas Papernot[2], Ian Goodfellow[3], Yan Duan[1,3], Pieter Abbeel[1,3]

[1]UC Berkeley, [2]Penn State, [3]OpenAI

Motivation

Adversarial example: small worst-case perturbation that forces a machine learning model to mishandle an input

- exists for image classification [1] and in the real world [2]



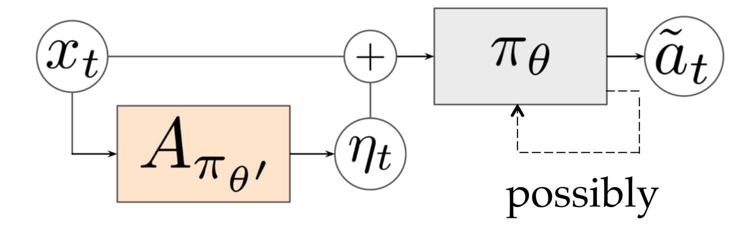
Key finding: Adversaries can degrade test-time behavior of policies trained with reinforcement learning, even when they do not have access to the policies

Threat Model

white-box adversary: $\theta' = \theta$

black-box adversary:

- adversary trains its own policy $\pi_{\theta'}$ for the task
- requires *transferability*: can adversarial ex. designed to fool one policy also fool others trained for the same task?
- challenging because training data drawn from different distributions:



supervised learning

For iterations $i = 1, \ldots, N$:
sample batch from $\mathcal{D}$
↓
update $\theta^{(i-1)}$ to $\theta^{(i)}$

$\mathcal{D}$ ground-truth (x, y) pairs

reinforcement learning

For iterations $i = 1, \ldots, N$:
sample rollouts from policy $\pi_{\theta(i-1)}$
↓
update $\theta^{(i-1)}$ to $\theta^{(i)}$

Dormant Adversarial Examples

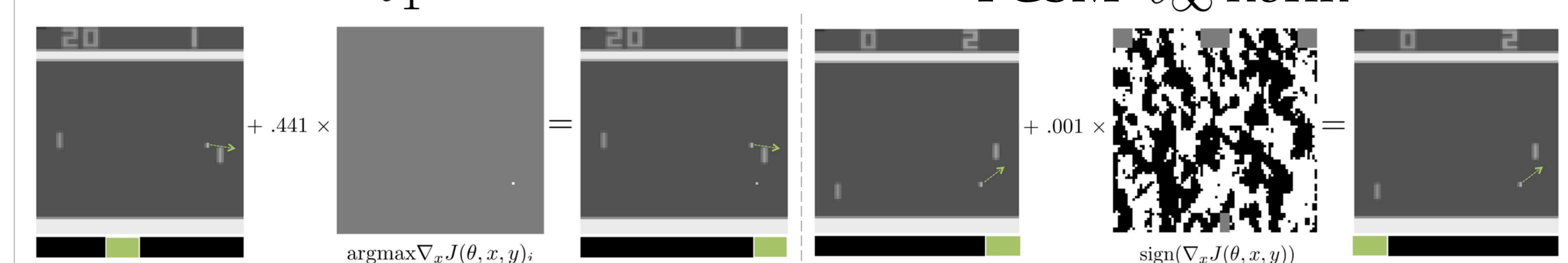
We introduce dormant attacks (on recurrent policies):

Time:	t	t+1	...	t+k-1	t+k
w/o adversary	✔	✔	...	✔	✔
w/ adversary	☒	☒	...	☒	☒
w/ *dormant* adversary	☑	✔	...	✔	✖

□ perturbation
✔ optimal
✖ suboptimal

Examples of Adversarial Perturbations

FGSM ℓ_1-norm FGSM ℓ_∞-norm



action taken: **up** original input · action taken: **down** adversarial input · action taken: **down** original input · inputs identical in 8-bit image space · action taken: **noop** adversarial input

Crafting Adversarial Examples for Policies

Optimal perturbation η, given loss $J(x)$: $\underset{\eta}{\text{argmax}}\ J(x+\eta)$

$$J(\theta, x, y) = -\sum_i y_i \log \pi_\theta(x)_i = -\log \underset{i}{\text{argmax}}\ \pi_\theta(x)_i$$

Fast gradient sign method (FGSM) [3] computes the optimal η for the linear approximation of $J(x)$

Original version of FGSM constrains $\|\eta\|_\infty$
Instead, we might want to constrain sparsity / magnitude

$$\eta = \begin{cases} \epsilon\ \text{sign}(\nabla_x J(\theta, x, y)) & \text{for } \|\eta\|_\infty \le \epsilon \\ \epsilon\sqrt{d}\ \frac{\nabla_x J(\theta,x,y)}{\|\nabla_x J(\theta,x,y)\|_2} & \text{for } \|\eta\|_2 \le \|\epsilon \mathbf{1}_d\|_2 \\ \text{maximally perturb dimensions with budget } \epsilon d & \text{for } \|\eta\|_1 \le \|\epsilon \mathbf{1}_d\|_1 \end{cases}$$

Crafting Dormant Adversarial Examples

$$\underset{\eta}{\text{argmax}} \quad J(\theta, x_t + \eta, y_t)$$

$$\text{subject to} \quad d_i(\eta; \theta, x_{0:t}) = 0 \ \text{ for } \ i = 0, \ldots, k-1,$$
$$d_k(\eta; \theta, x_{0:t}) = 1, \ \|\eta\| \le \epsilon$$

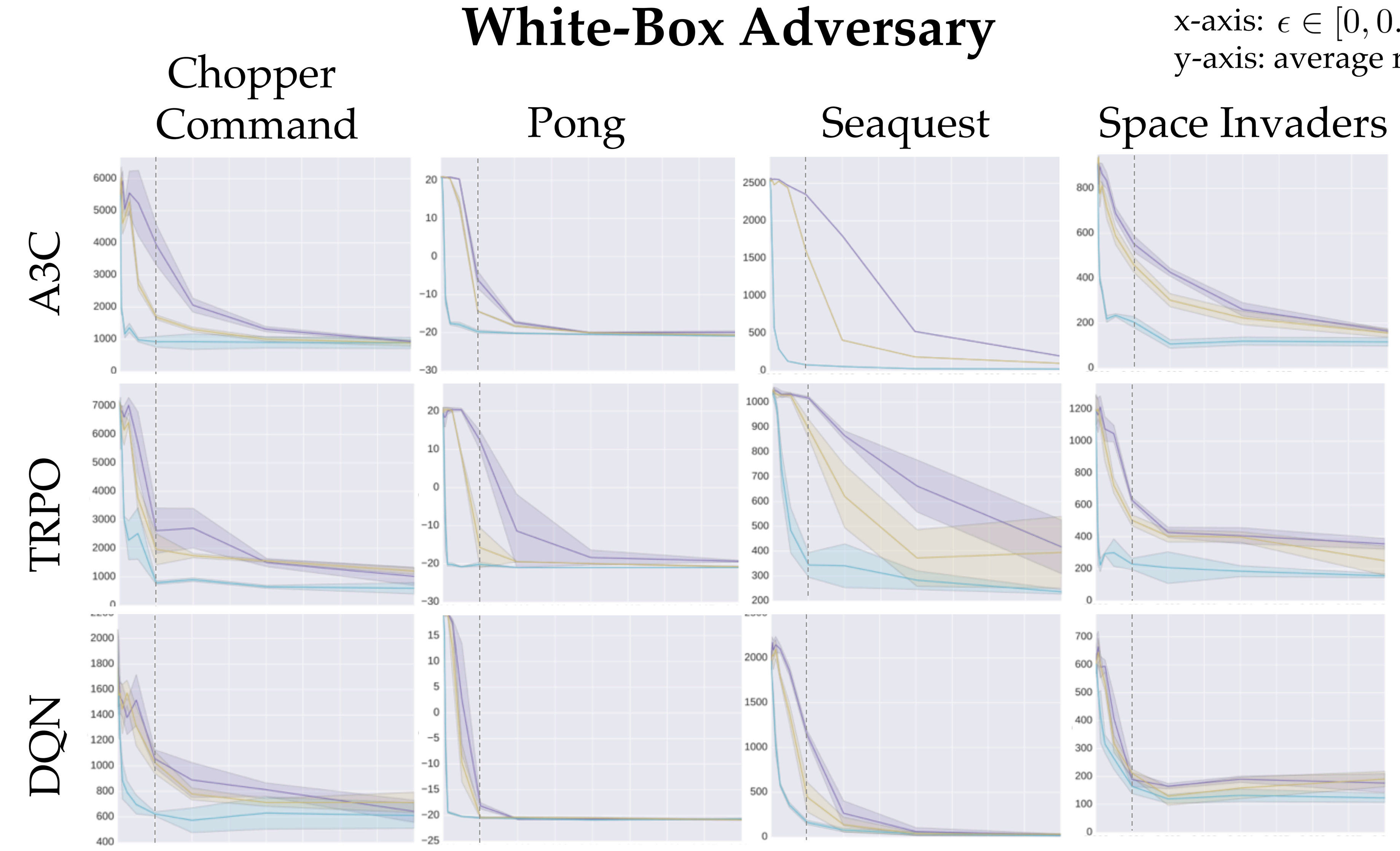
Optimize with dual ascent:

$$\eta^{(j)} = -\epsilon\left(\sum_{i=0}^{k-1} \lambda_i^{(j)}\ \text{sign}(\nabla_x J(\theta, x_{t+i}, y_{t+i}))\right) + \epsilon\ \lambda_k^{(j)}\ \text{sign}(\nabla_x J(\theta, x_{t+k}, y_{t+k})).$$

$$\lambda_i^{(j+1)} = \lambda_i^{(j)} + \alpha^{(j)}\ d_i(\eta^{(j)}; \theta, x_{0:t}) \ \text{ for } \ i = 0, \ldots, k$$

White-Box Adversary

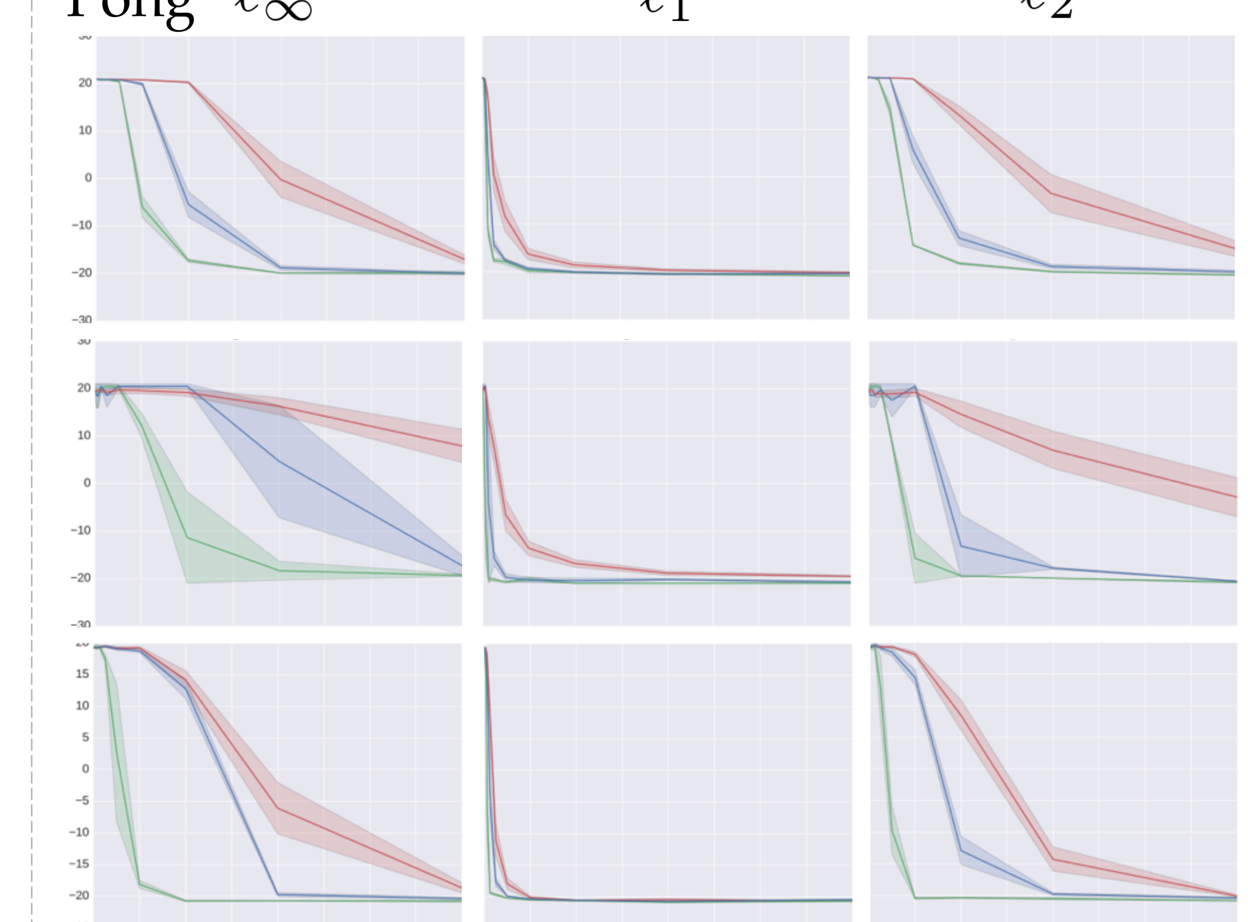
x-axis: $\epsilon \in [0, 0.008]$
y-axis: average return



Type of perturbation: ℓ_∞ ℓ_2 ℓ_1

- Across all games, adversarial perturbations significantly decrease performance, even for small ϵ
- Policies trained with DQN tend to be more vulnerable

Black-Box Adversary



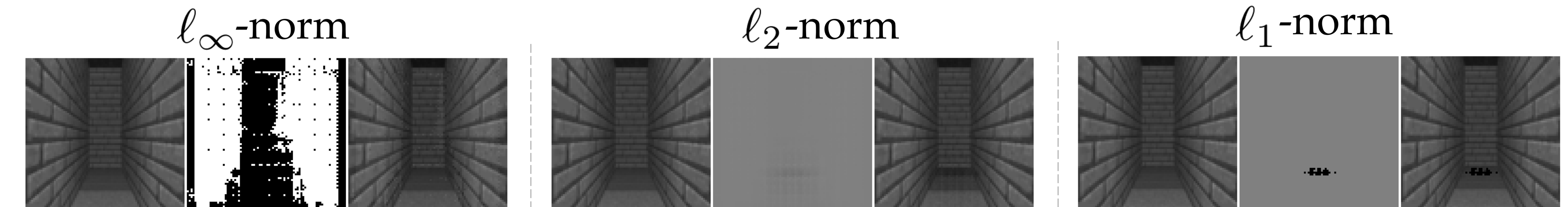
Type of transfer: algorithm policy none

- ℓ_1-norm is particularly transferable, even across training algorithms

Dormant Adversary

Task: Navigate I-maze [4]

dormant adversarial examples computed through dual-ascent (k = 7)



[1] Szegedy et al. Intriguing properties of neural networks. ICLR 2014
[2] Kurakin et al. Adversarial examples in the physical world. arXiv 2016.
[3] Goodfellow et al. Explaining and harnessing adversarial examples. ICLR 2015.
[4] Oh et al. Control of memory, active perception, and action in Minecraft. ICML 2016.
[5] Kempka et al. ViZDoom. arXiv 2016.