Adversarial examples in RL

Clément Acher
Hugo Cisneros

January 30th, 2019

Outline

I - Adversarial Examples in Deep Learning

II - Adversarial Examples in Deep Reinforcement Learning

a) Attacks
b) Defense

Outline

I - Adversarial Examples in Deep Learning

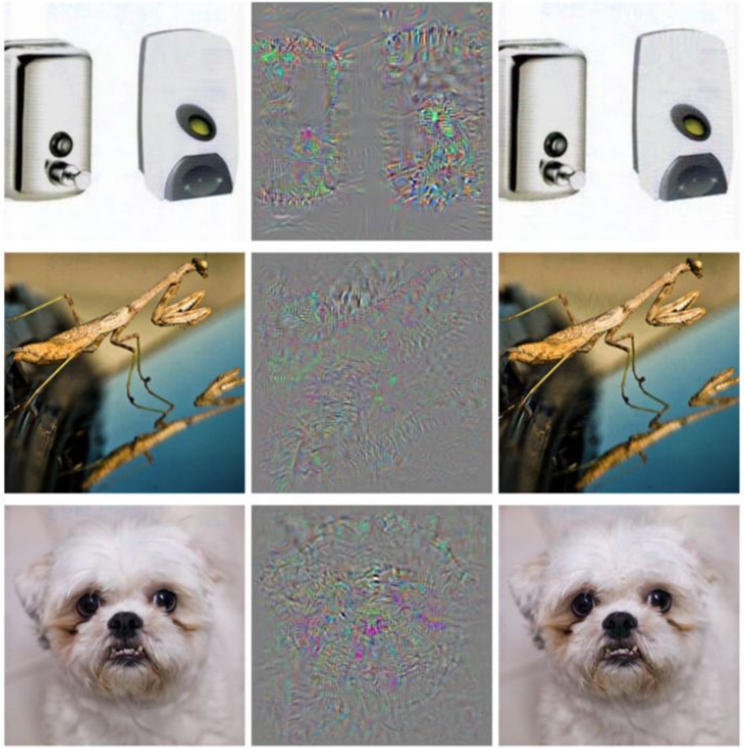
II - Adversarial Examples in Deep Reinforcement Learning

a) Attacks
b) Defense

Adversarial Examples in DL

Adversarial Examples: instance with small, intentional feature perturbations that causes a model to make a false prediction.

2013 : "Intriguing properties of neural networks" - Szegedy et al : DL models are vulnerable



Correctly classified

Classified as *ostrich*

Adversarial Examples in DL

- Does the attacker have the access to the trained model?

Yes → *Whitebox* attack

No → *Blackbox* attack
Papernot et al. (2016)

- Does the attack have the goal to have the model predict a specific target?

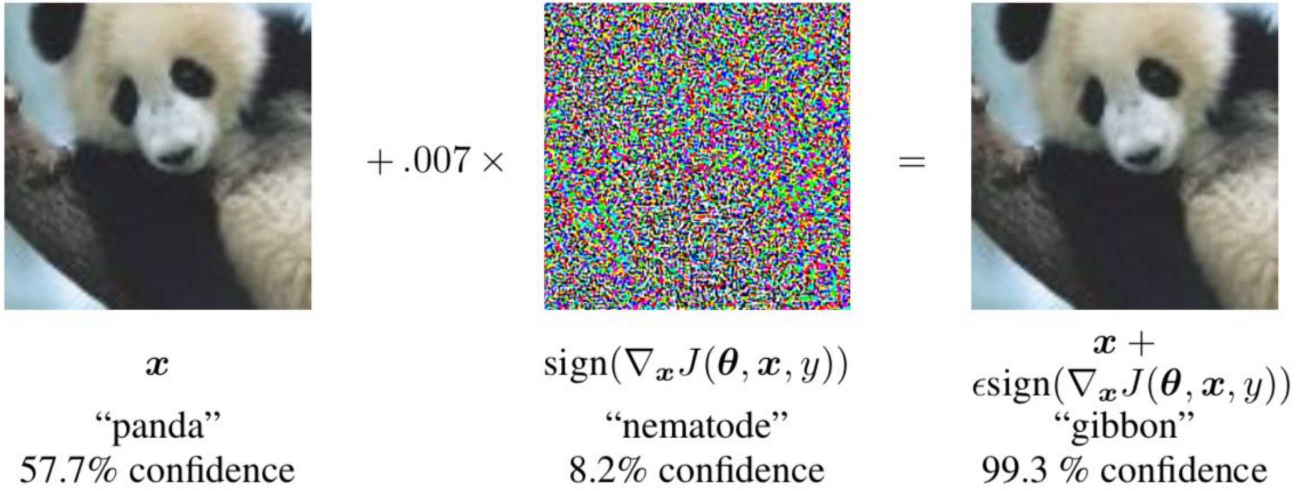
Yes → *Targeted* attack

No → *Non-targeted* attack

Adversarial Examples in DL - Crafting an attack

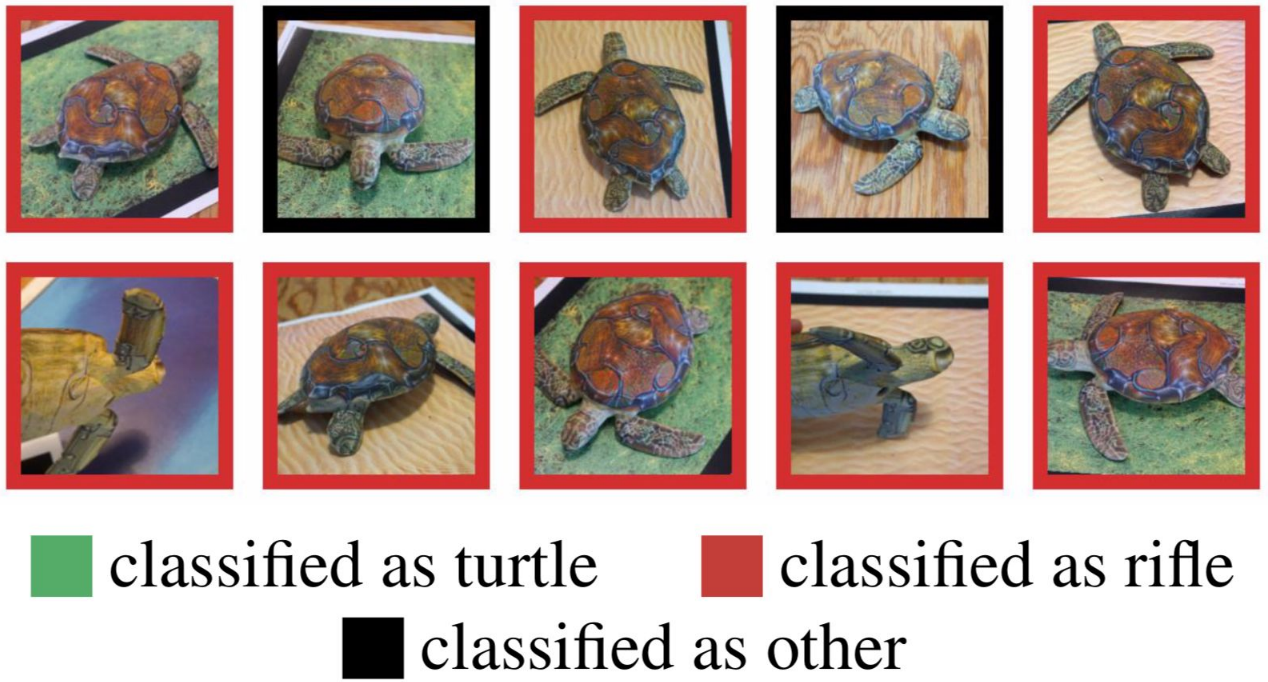
Fast Gradient Sign Method (FGSM) : untargeted attack in a whitebox setting

Assumption that loss *J* is linear around the input *x*



Goodfellow et al (2014)

Some highlights - 3D printed adversarial examples



Synthesizing Robust Adversarial Examples, Athalye et al. (2017)

Outline

I - Adversarial Examples in Deep Learning

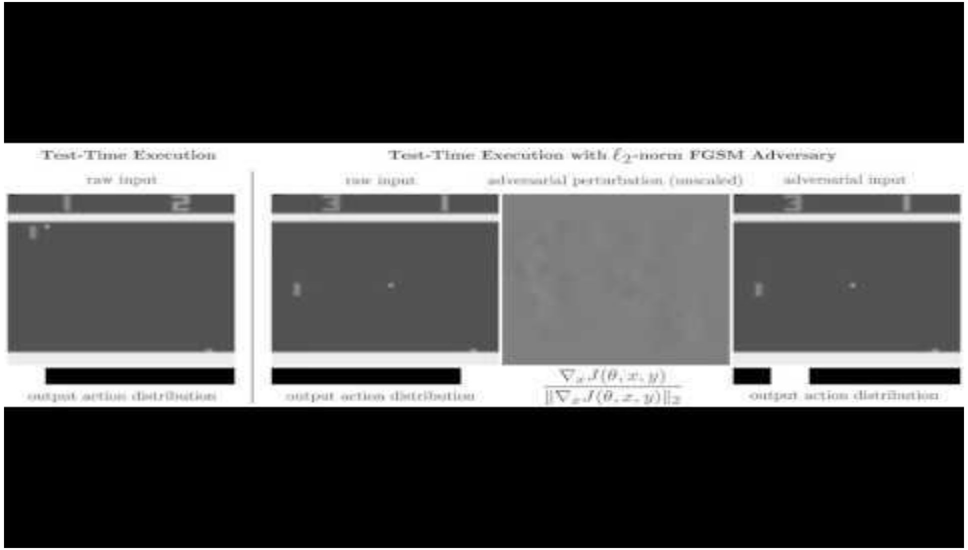
II - Adversarial Examples in Deep Reinforcement Learning

a) **Attacks**
b) Defense

Attacks in DRL - First proofs of vulnerability

Behzadan and Munir (2017) + Huang et al. (2017)

- DQN/TRPO/A3C
- MITM attack based on FGSM/JSMA methods
- Both blackbox and whitebox settings



Attacks in DRL - Tailored attacks for DRL

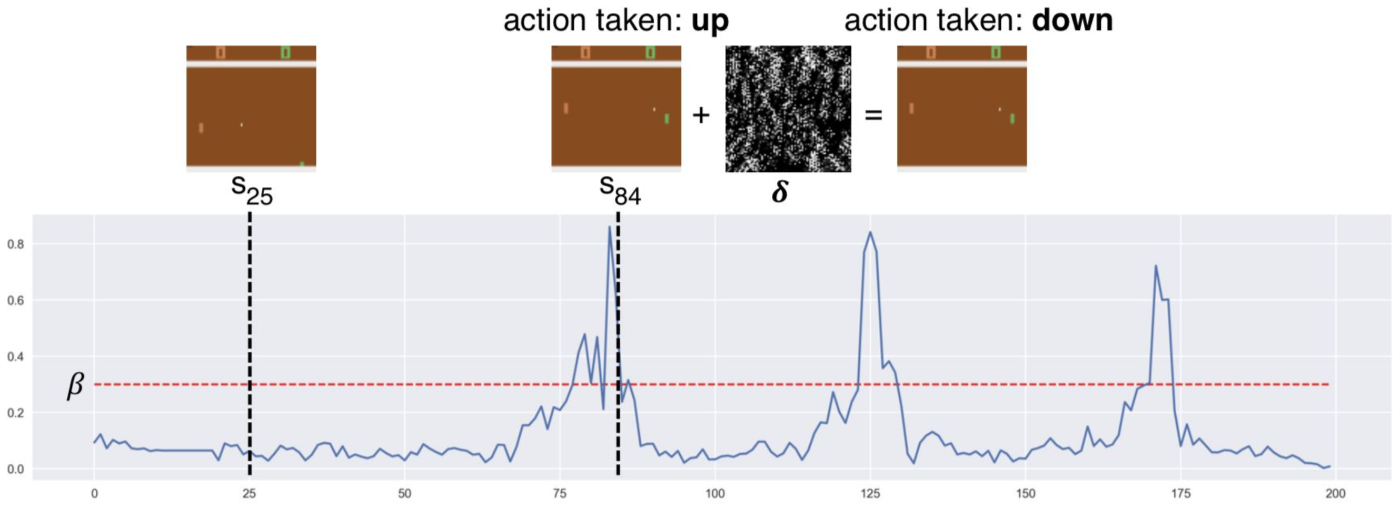
Pattanaik et al. (2017) : adversarial examples should force the agent to perform the *worse* action

Use a different loss :

$$J(s, \pi^*) = -\sum_{i=1}^{n} P(a_i) \log \pi_i^* = -\log \pi_w^*$$

Where *w* corresponds to the worst action

Attacks in DRL - Strategically-timed attacks



Lin et al. : Performs attacks only on selective time steps

$$c\left(s_t\right) = \max_{a_t} \pi\left(s_t, a_t\right) - \min_{a_t} \pi\left(s_t, a_t\right)$$

Attacks in DRL - Targeted attacks

- Lin et al. : Lure the agent toward an adversarial <u>state</u>
  <u>Technique:</u>
  1. Plan the sequence of action needed to reach the malicious state
  2. Craft the successive adversarial examples

  <u>Results:</u> 70% success rate in 3 out of 5 games

- Trestschk et al. : Lure the agent to follow an adversarial <u>reward</u>
  <u>Technique:</u> Use a neural network g_theta to craft the adversarial examples.
  $x \mapsto Q(x + g_\theta(x))$ is trained as a DQN
  <u>Results:</u> attacked agent behaves similarly as the agent trained on the adversarial reward

Outline

I - Adversarial Examples in Deep Learning

II - Adversarial Examples in Deep Reinforcement Learning

a) Attacks
b) Defense

Defense in DRL

Research into building defense for deep RL models has been based on several approaches

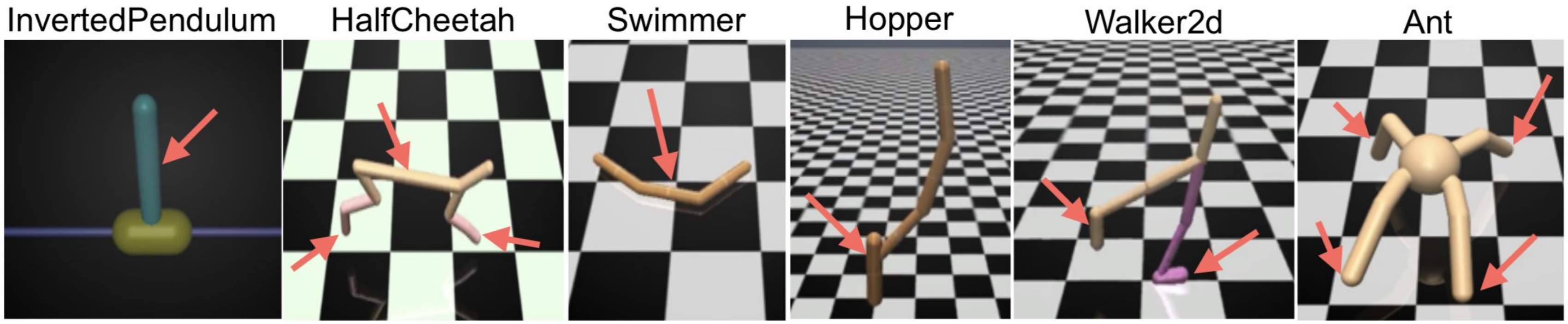
- Adversarial training and extensions
- Predictive defense
- Meta-learning
- Noisy exploration

Defense in DRL - Adversarial training

Train the model in an **adversarial environment**.

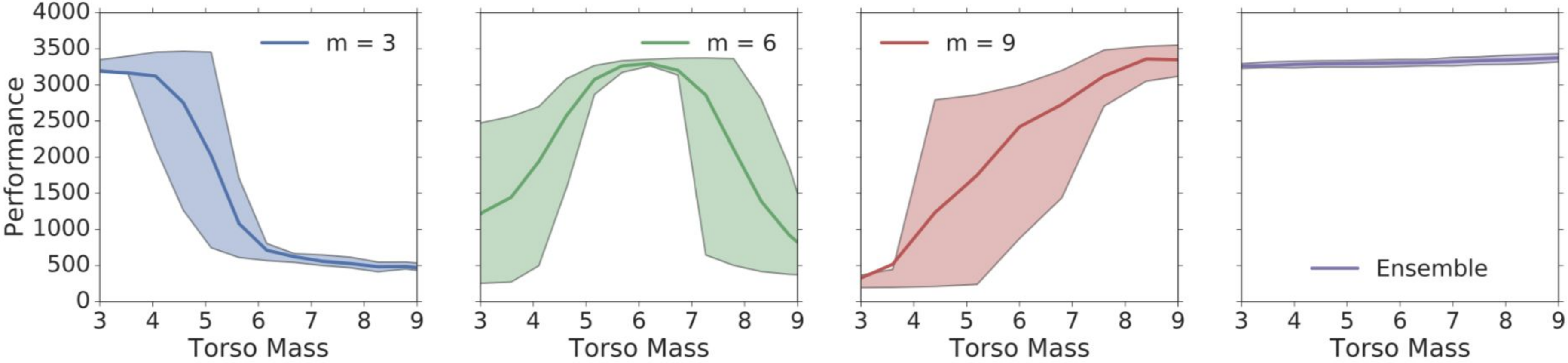
How to generate a generic adversarial setting that an agent can learn from ?

- Morimoto et al.: Robust RL, **Min-max problem** with two opposite control
- Pinto et al.: Extend the idea by having two agents learn the perturbations with **opposite rewards**. Apply to a deep RL setting.



Defense in DRL - EPOpt

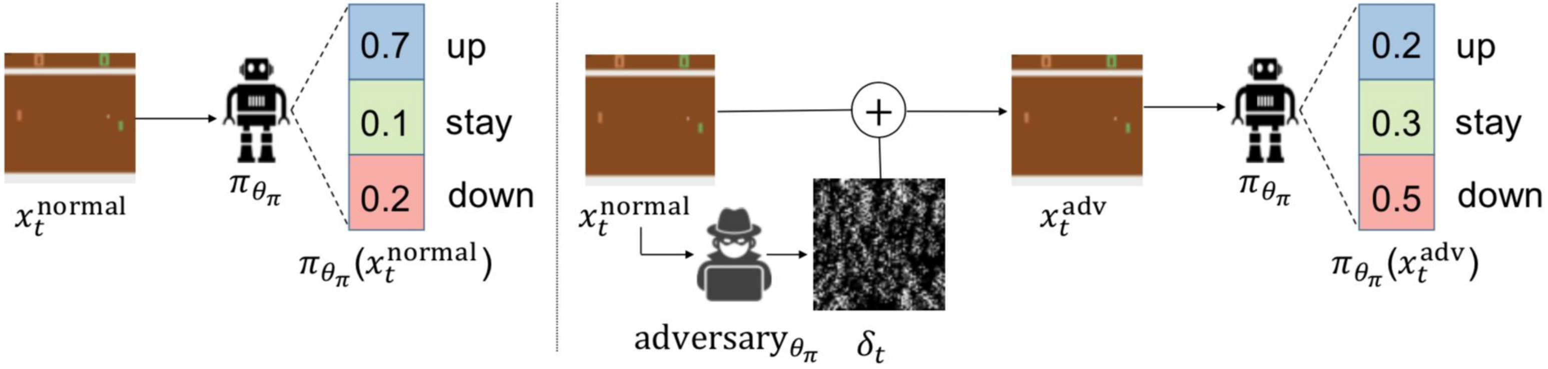
Rajeswaran et al.: Similarly to adversarial training, sample from *ensemble MDP* parameters and train **on worst ε-percentile** trajectories**.**



It results in policies that generalize well to a range of model parameters and are therefore robust to adversarial examples affecting them.

Defense in DRL - Predictive defense

Lin et al.: Predict the “normal” next state to use as indicator of an attack.



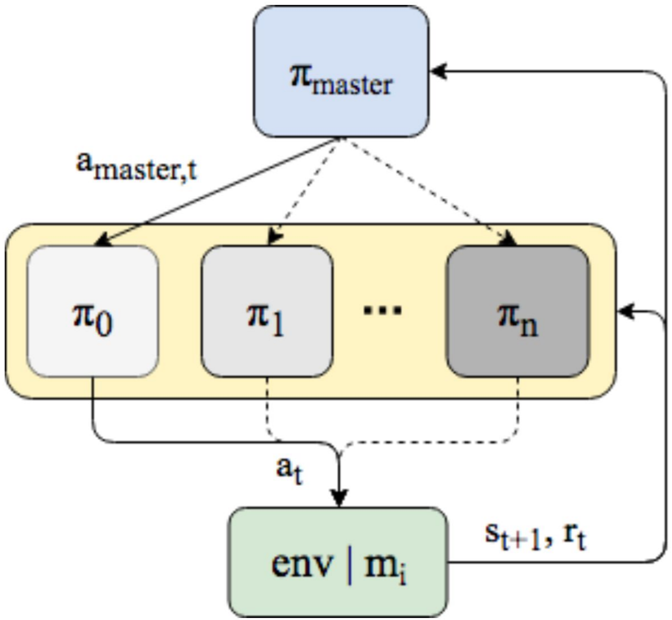
(a) *without* adversary (b) *with* adversary

Distance between action space distributions is used as a trigger.

Defense in DRL - Meta-learning

Havens et al.: A master policy is trained to recognize if the agent is being attacked.

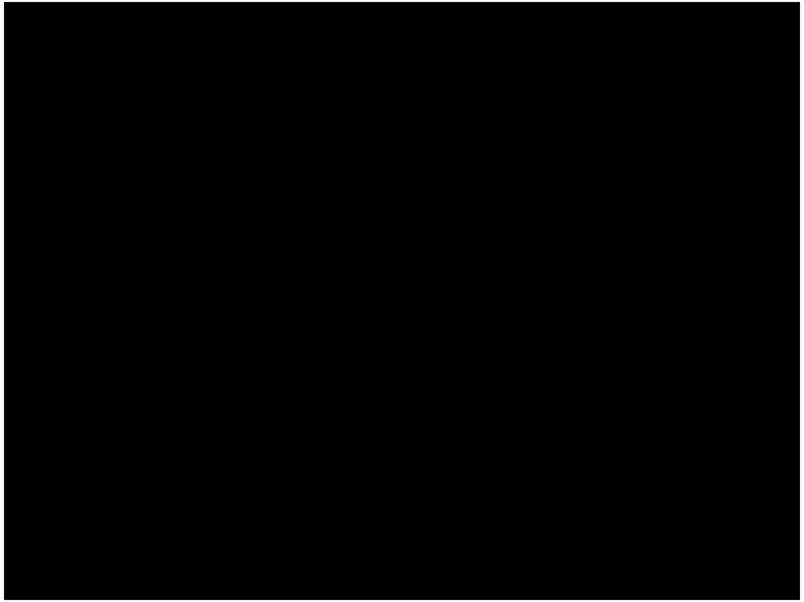
It switches at each step to a policy that has either been trained on unperturbed data or is learned on the fly



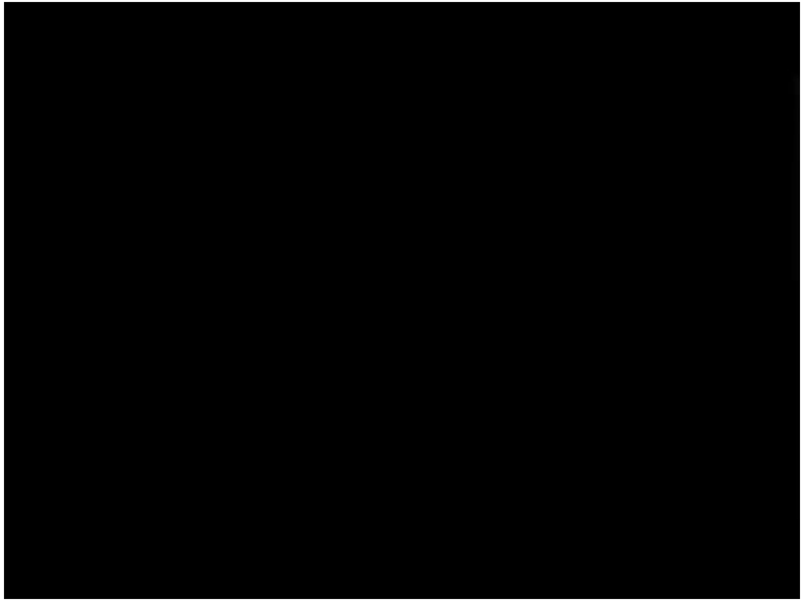
Discussion

- Attacks only performed on high-dimension spaces
- No real world attack
- Attacks target the state as perceived by the agent: other means (reward) could be targeted

Some highlights - Hidden voice commands



Whitebox setting



Blackbox setting

Hidden Voice Commands, Carlini et al. (2016)