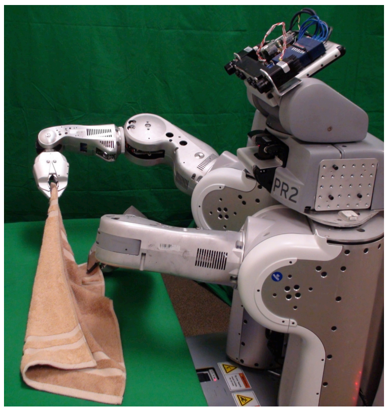
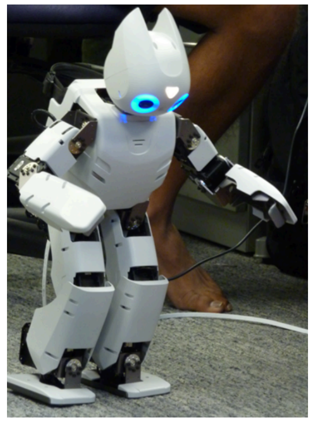
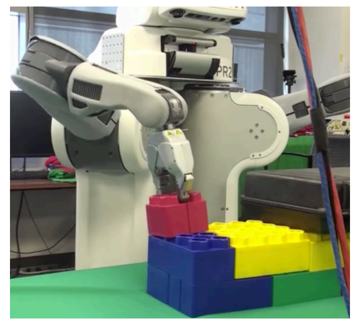
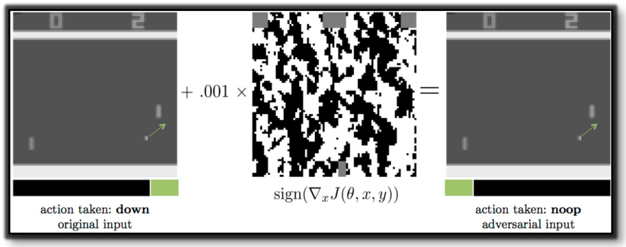


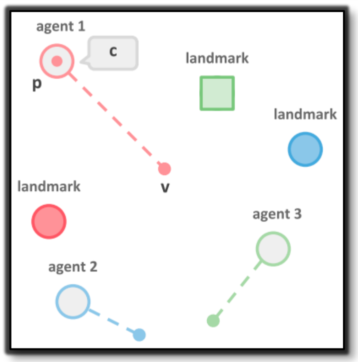



CS294-112 SP17 Guest Lecture

Pieter Abbeel

Outline





1) Adversarial Attacks on Neural Network Policies
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

2) Emergence of Grounded Compositional Language in Multi-Agent Populations
Igor Mordatch, Pieter Abbeel

3) Autonomous Helicopter Flight
Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng

Adversarial Examples in RL

- Can RL agents be brainwashed?
- Can RL agents be trained to be sleeper agents?

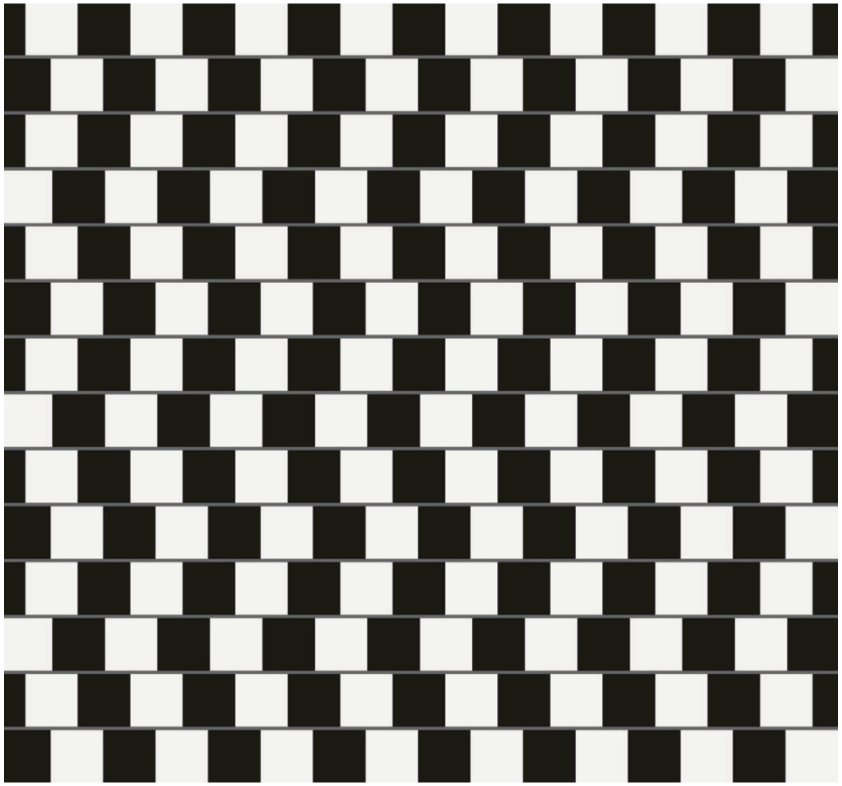
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Spot the differences

?

[slide from Papernot]

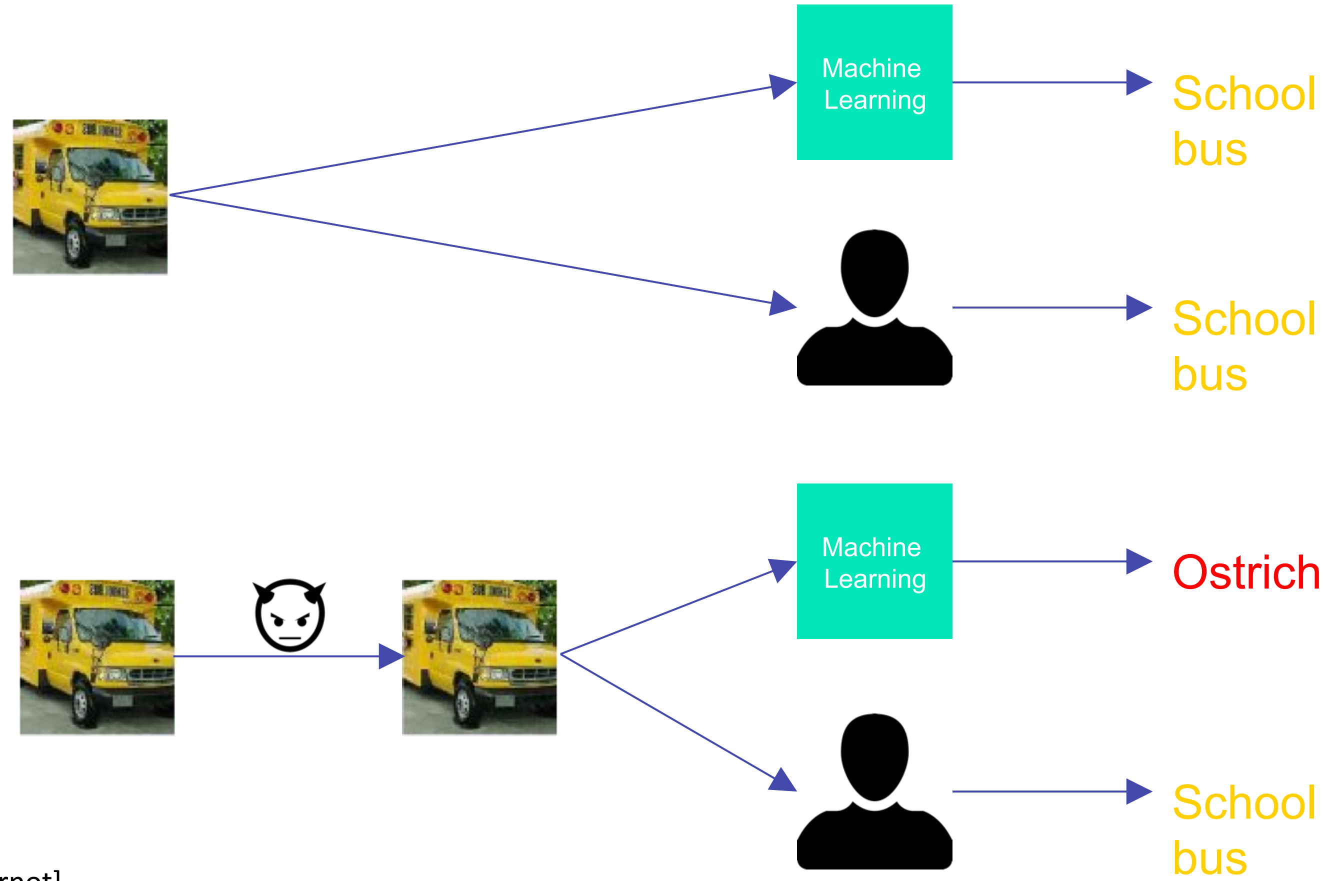
Humans can be fooled too !



http://i.imgur.com/TTpIGvo.jpg
http://www.wired.com/wp-content/uploads/2015/10/Cofeehouse-%C2%AEThomas_Hunt-1024x957.jpg

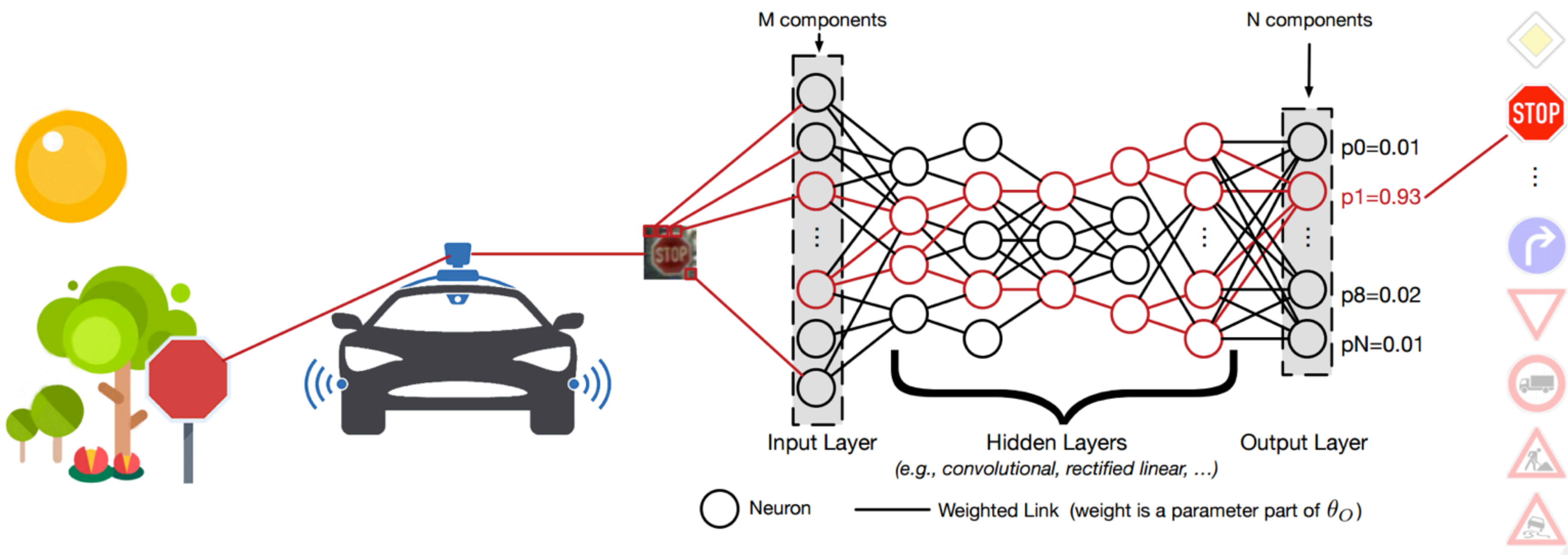
[slide from Papernot]

Adversarial Examples



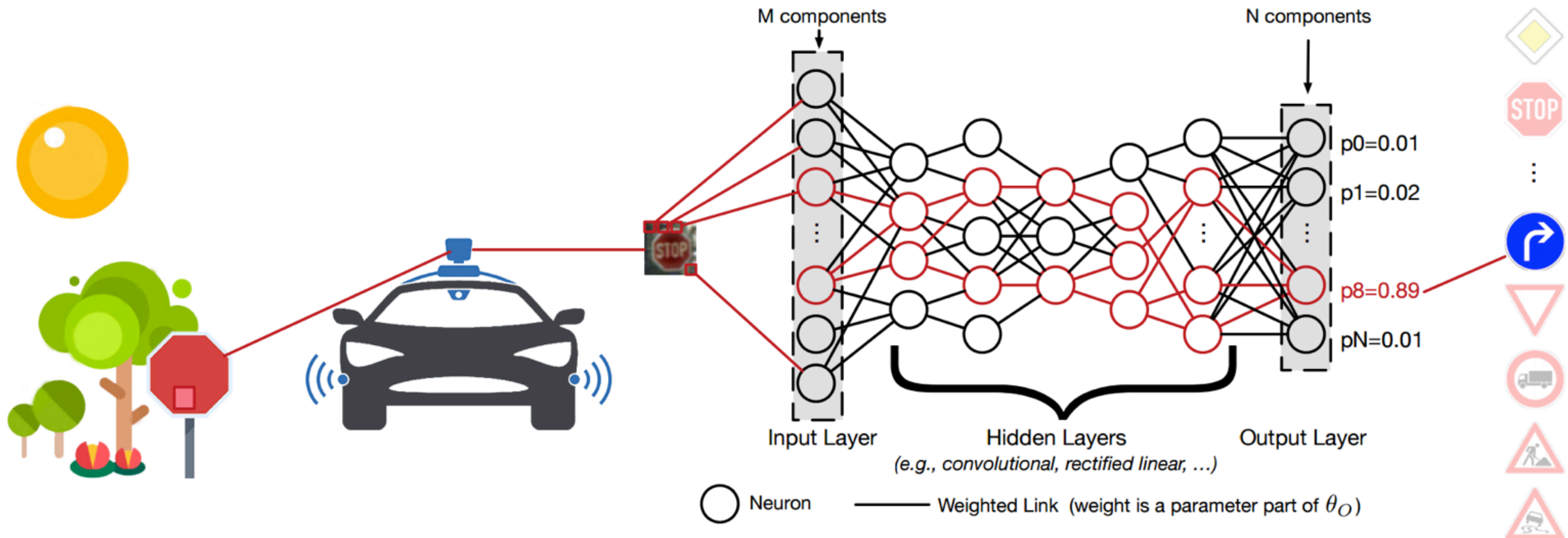
[slide from Papernot]

Adversarial Examples



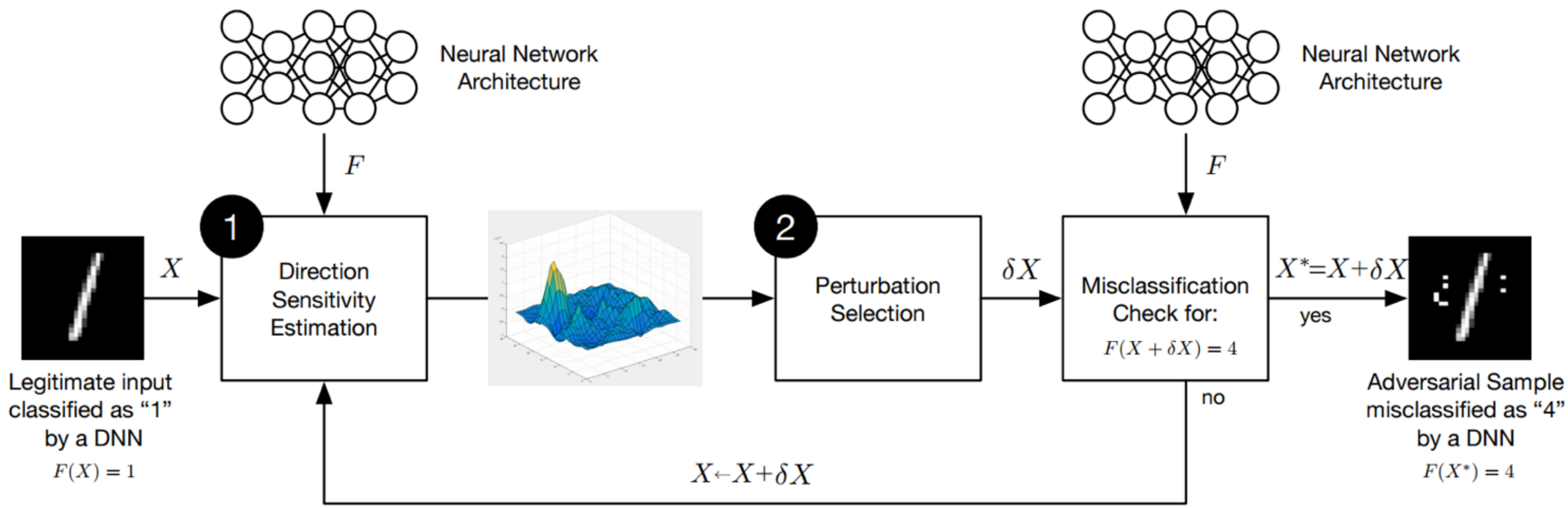
[slide from Papernot]

Adversarial Examples



[slide from Papernot]

Jacobian-Based Iterative Approach: *source-target misclassification*



[PMJ16] Papernot et al. The Limitations of Deep Learning in Adversarial Settings

Jacobian-Based Iterative Approach: *source-target misclassification*

Source-target attack on MNIST (test set)

97.05%	adversarial success rate
4.03%	average distortion

Source-target attack on CIFAR-10 (test set)

92.78%	adversarial success rate

If only interested in **misclassification**

MNIST	1.55% average distortion
CIFAR-10	0.39% average distortion

[PMJ16] Papernot et al. The Limitations of Deep Learning in Adversarial Settings
[PMW16] Papernot et al. Distillation as a Defense against Adversarial Perturbation of Deep Neural Networks

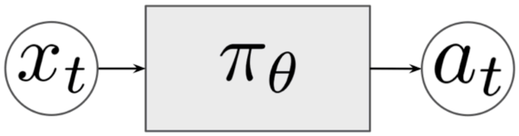
Adversarial Examples in RL

- Can RL agents be brainwashed?
- Can RL agents be trained to be sleeper agents?

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Threat Model

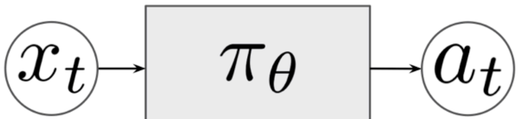
No adversary



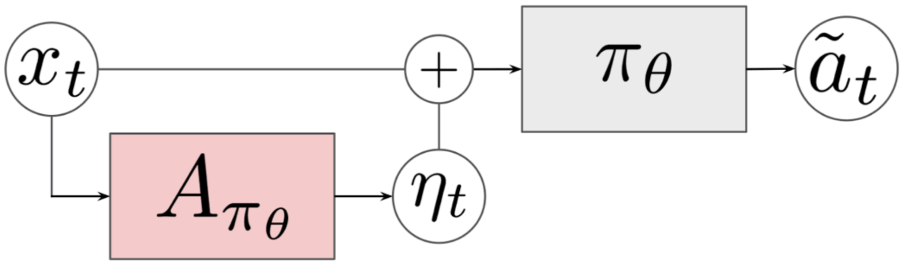
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Threat Model

No adversary

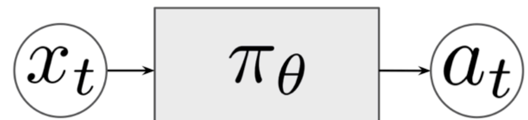


White-box adversary

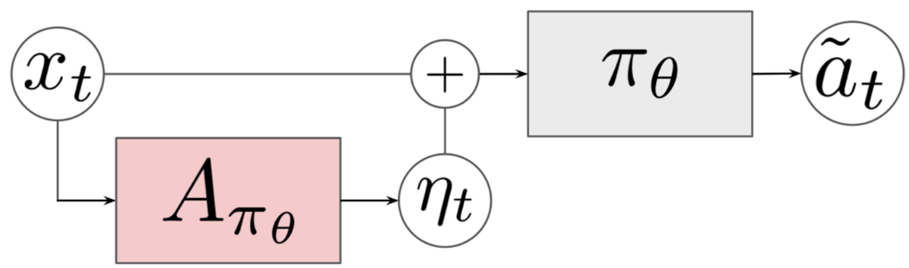


Threat Model

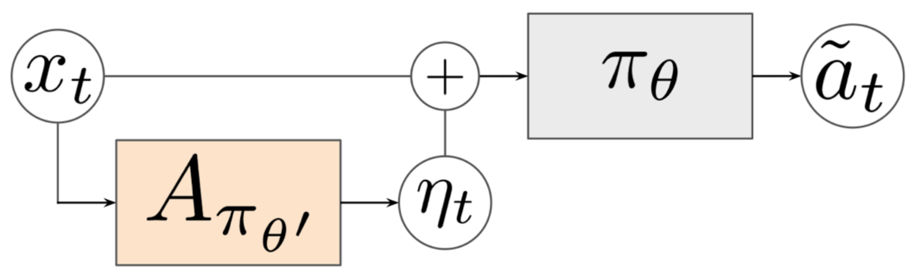
No adversary



White-box adversary



Black-box adversary



Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Adversarial Example Crafting

Adversarial example: $\tilde{x} = x + \eta$

Optimal adversarial perturbation η, given loss function $J(x)$:

$$\underset{\eta}{\operatorname{argmax}}\ J(\tilde{x})$$

Adversarial Example Crafting

Adversarial example: $\tilde{x} = x + \eta$

Optimal adversarial perturbation η, given loss function $J(x)$:

$$\underset{\eta}{\operatorname{argmax}} \; J(\tilde{x})$$

Fast gradient sign method[1] (FGSM) computes the optimal η for the linear approximation of $J(x)$, under the constraint $\|\eta\|_\infty \leq \epsilon$:

$$\eta = \epsilon \; \operatorname{sign}(\nabla_x J(x))$$

- efficient, reliably fools image classifiers

[1]Goodfellow et al., ICLR 2015

Norm Constraints for FGSM

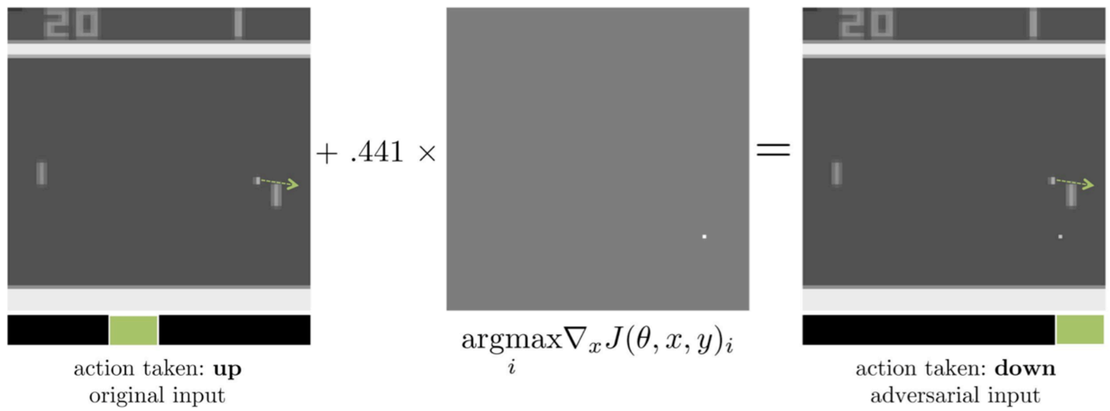
Original version of FGSM constrains $\|\eta\|_\infty$

Instead, we might want to constrain the sparsity or magnitude of η

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

Examples

FGSM
ℓ_1 norm

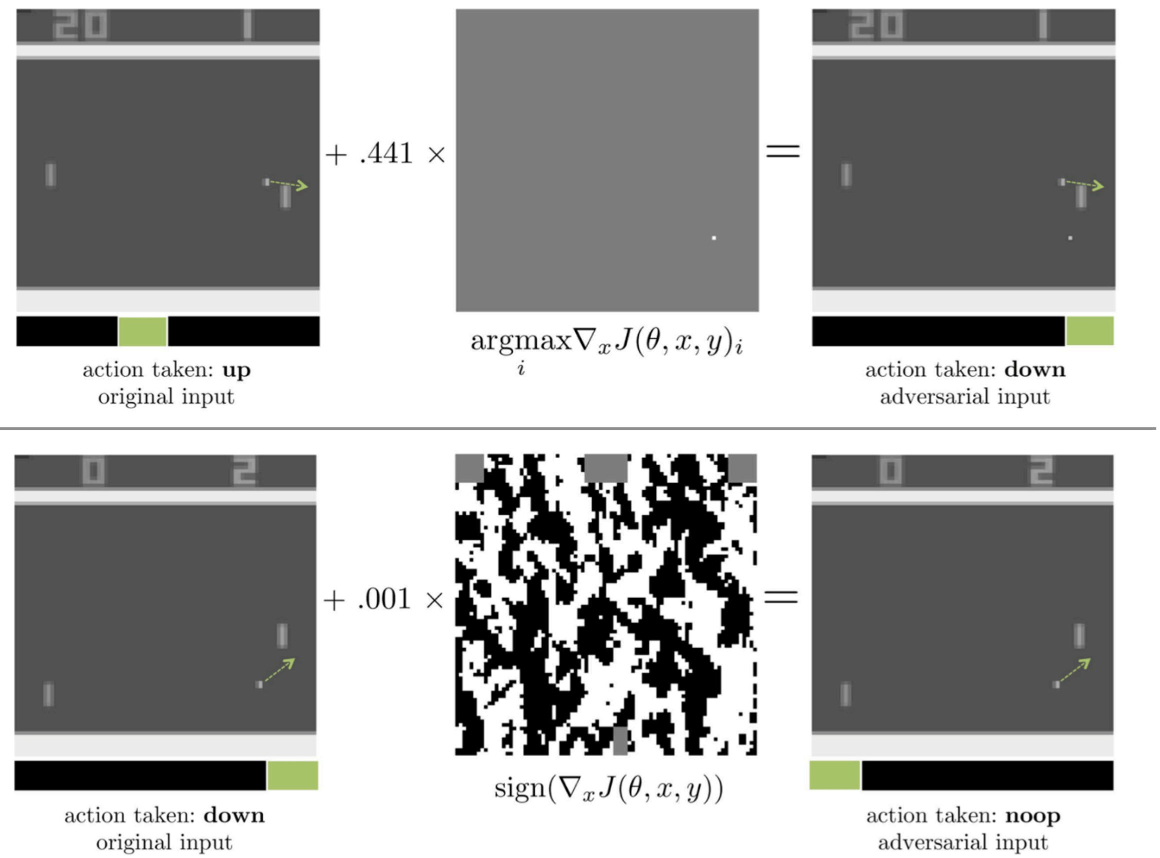


Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Examples

FGSM
ℓ_1 norm

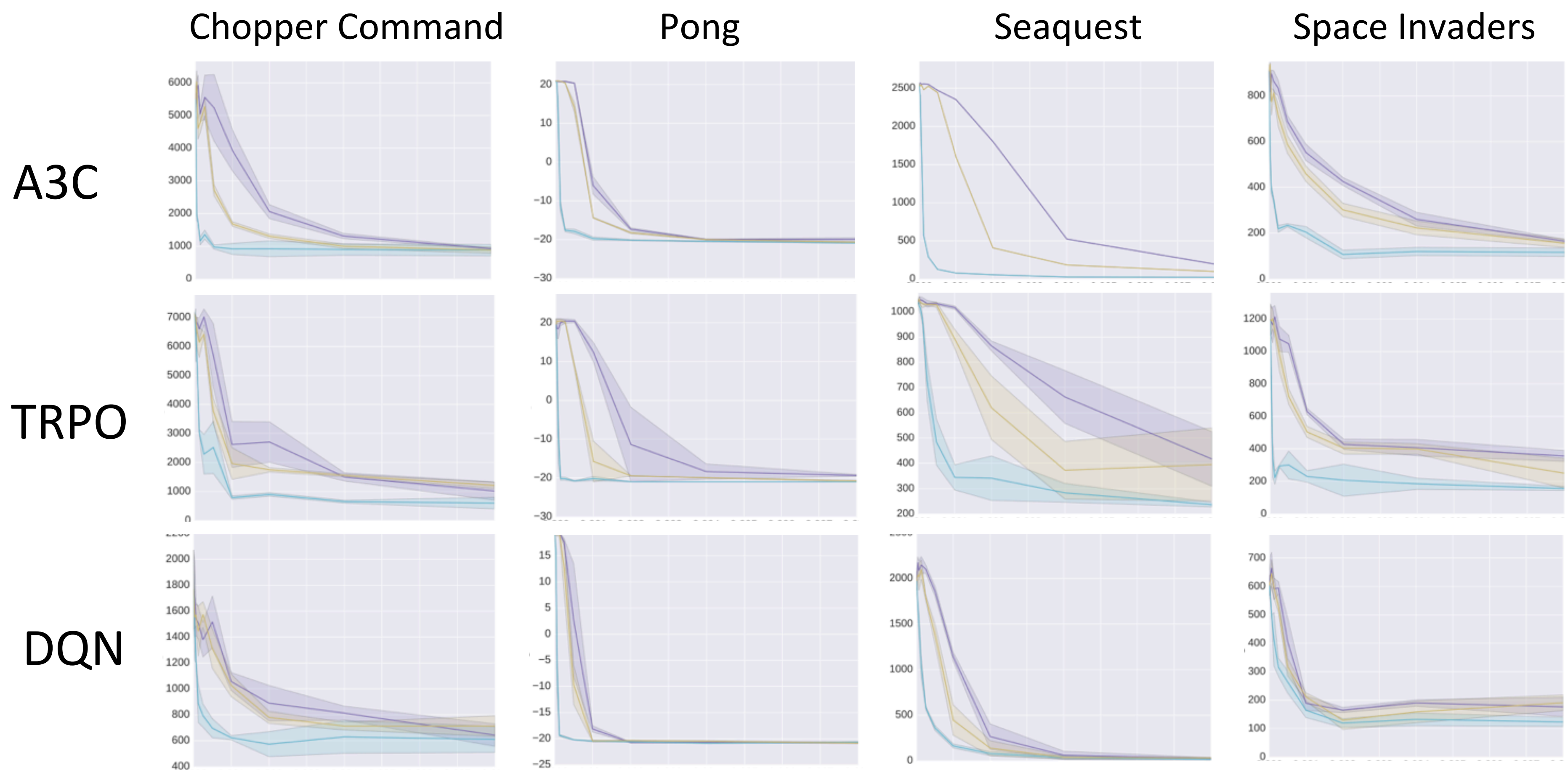
FGSM
ℓ_∞ norm



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

Results: White-Box

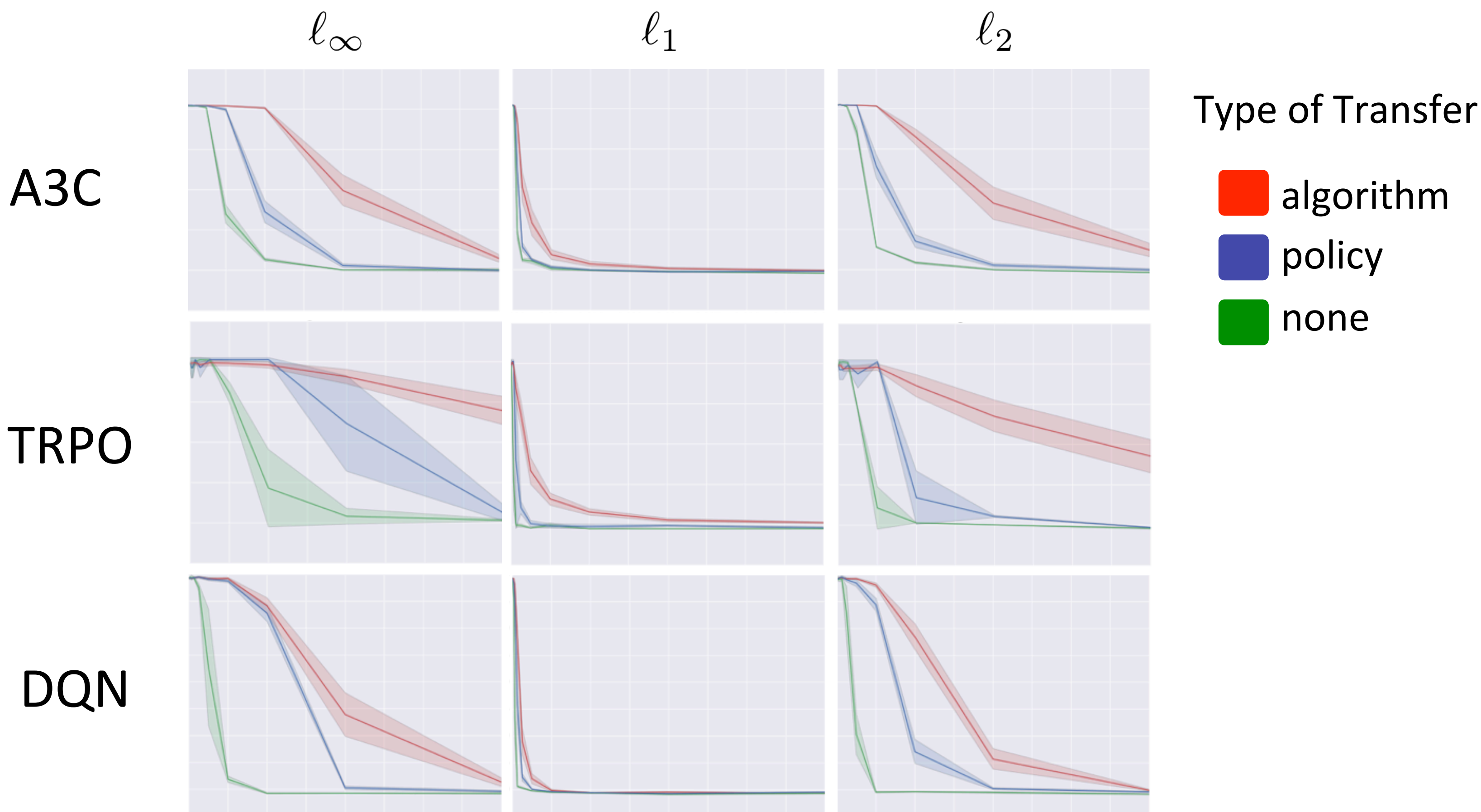
ℓ_∞ ℓ_2 ℓ_1



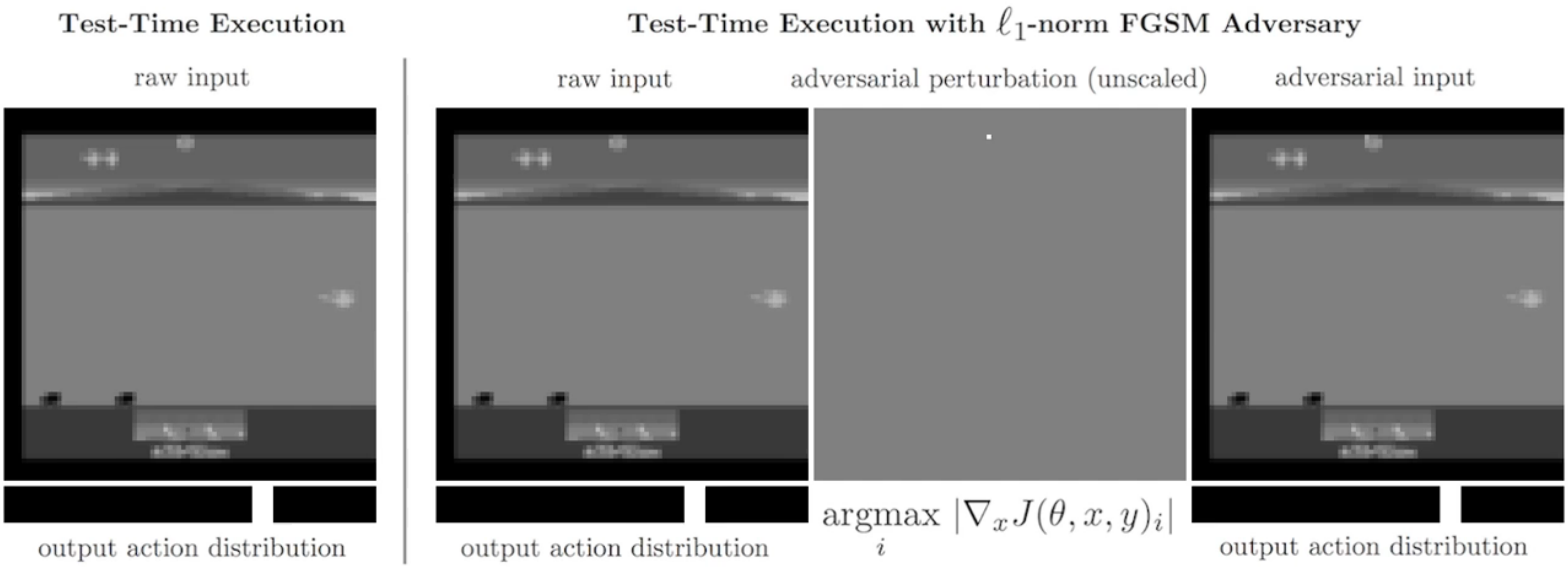
x-axis:
y-axis: average return

Results: Black-Box

Pong



Results: Black-Box



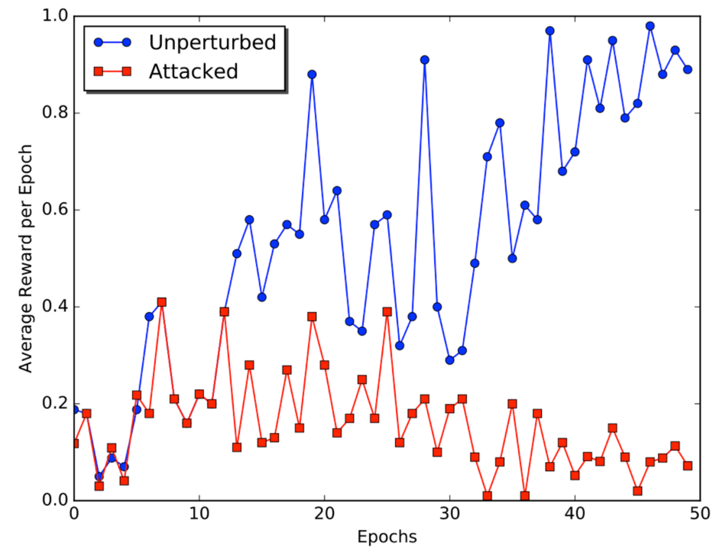
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Related Work

"Vulnerability of Deep RL to Policy Induction Attacks"

Behzadan & Munir
arXiv 2017

Goal: prevent policy from learning how to optimize true reward r



Approach:

1. adversary trains policy to optimize $-r$
2. at every time step t, choose η_t to lead target policy to select same action as adversary's policy[1]

In addition, analyzes white- and black-box adversarial attacks on a fully trained policy at individual time steps (not across an entire policy rollout)

[1] uses JSMA to choose η_t [Papernot et al., EuroS&P 2016]

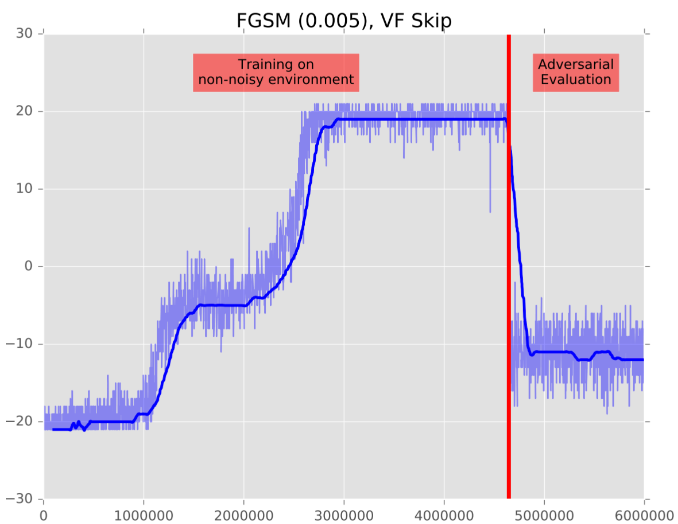
Related Work

"Delving into Adversarial Attacks on Deep Policies"

Kos & Song, ICLR 2017
workshop submission

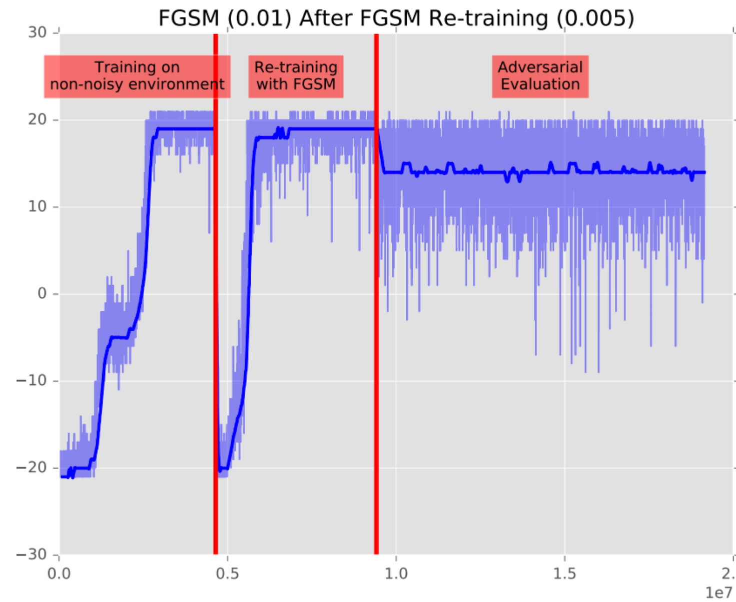
Goal 1: inject fewer perturbations

only perturb if value of state x_t exceeds threshold (≈10% of time steps)



Goal 2: defend against adversary

retrain on adversarial perturbations



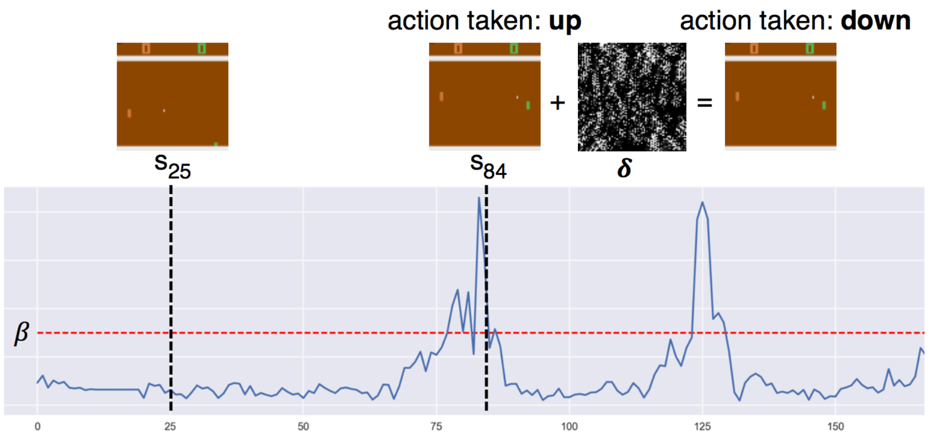
Related Work

"Tactics of Adversarial Attacks on Deep RL Agents"

Lin et al., ICLR 2017
workshop submission

Goal 1: inject fewer perturbations

only perturb if $\max(a_t) - \min(a_t)$ exceeds threshold (≈25% of time steps)



Goal 2: lead agent to state x_G

1. train video prediction model to predict x_{t+H}, given x_t and $a_{t:t+H-1}$
2. use cross-entropy method to find sequence of H actions to reach x_G
3. choose best perturbation at current time t, to lead agent to perform first action in sequence
4. repeat #2 and #3 until x_G is reached (i.e., use model predictive control)

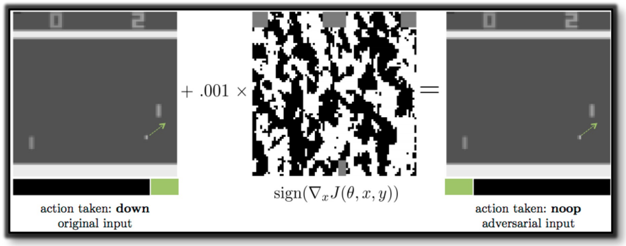
Current Directions

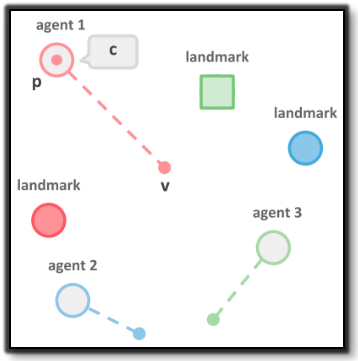
Adversarial-example attacks on memory-based policies
    dormant attacks: delayed negative effect
    memory-corrupting attacks: cause policy to forget its goal or task

Control agent to optimize a different reward function

Adversarial examples on neural network policies, in the real world

Outline



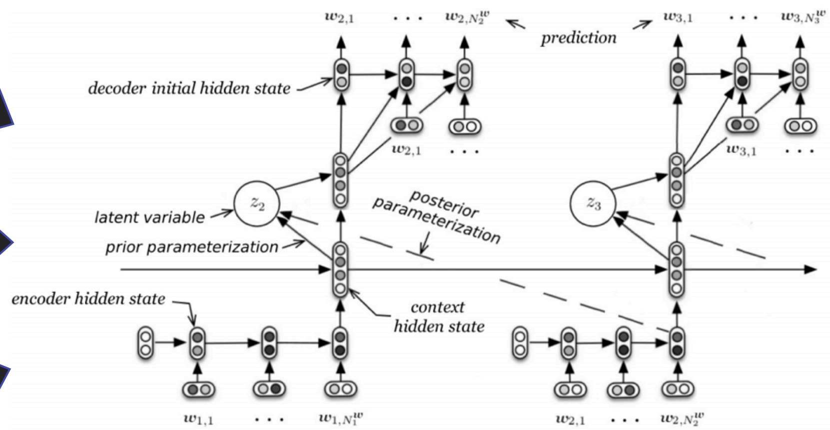


1) Adversarial Attacks on Neural Network Policies
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

2) Emergence of Grounded Compositional Language in Multi-Agent Populations
Igor Mordatch, Pieter Abbeel

3) Autonomous Helicopter Flight
Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng

Most Common Paradigm: Learning on Static Datasets



"Hi! How are you doing today?"

Most Common Paradigm: Learning on Static Datasets

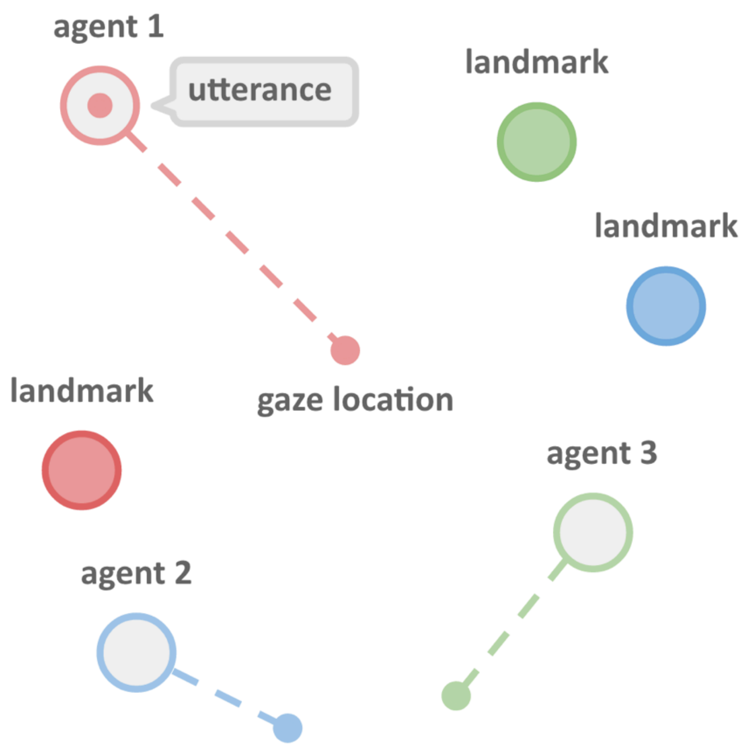
- Train deep neural networks on large, task-specific datasets using (mostly) supervised learning
- Has enabled many practical advances in machine translation (Bahdanau et al., 2014), sentiment analysis (Socher et al., 2013), document summarization (Durrett et al., 2016), dialogue (Dhingra et al., 2016)

Is there anything missing?

Grounding

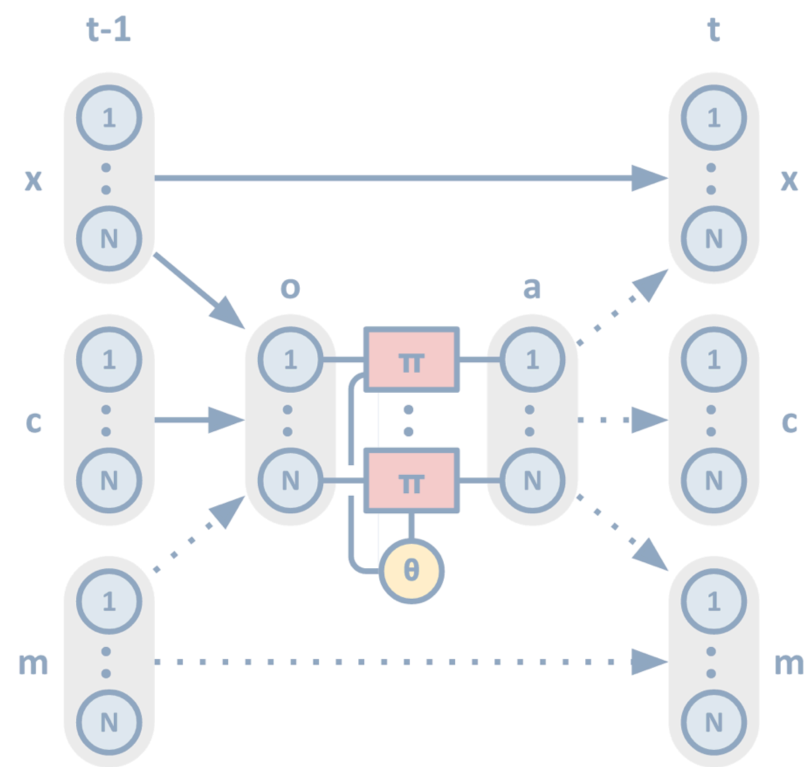
- Idea that words in a language are tied to something directly **experienced by a speaker in their environment**
- Deep learning on static datasets learns the **statistical structure** of language
- <u>But this may not be sufficient:</u> we want agents to understand language so they can **carry out real tasks** in the world (or on the Internet)

Multi-Agent Environments



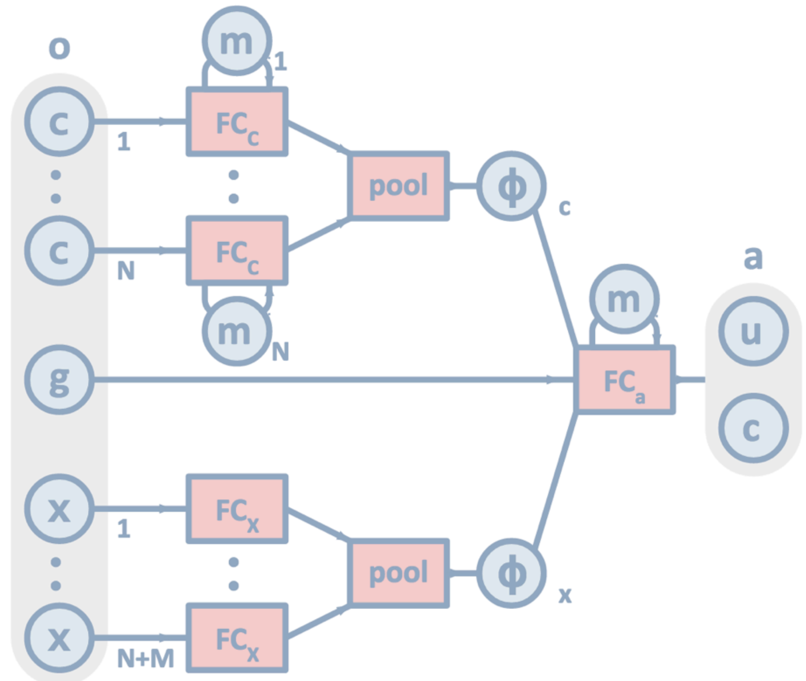
Multi-agent communication

- Communication outputs and environment actions are **discrete**
- Environment state is **continuous**
- Agents share parameters
- Communication symbols are abstract one-hot vectors



Agent policies

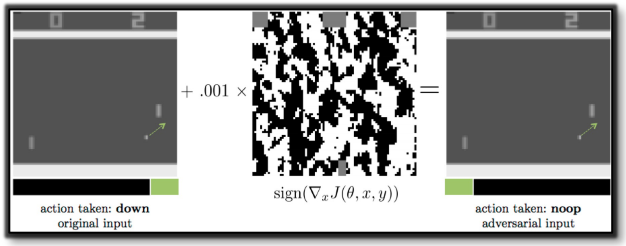
- Stochastic policies represented by **recurrent modules** with **memory**
- Trained end-to-end with backpropagation through time
- Use Gumbel-Softmax trick (Jang et al., 2016) for backpropagating through discrete actions

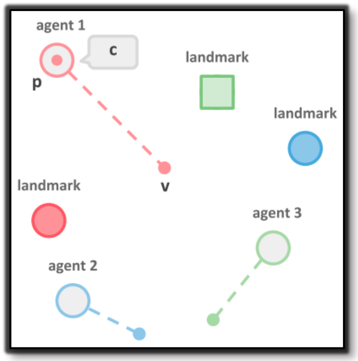


Compositional Communication

0; 1:43

Outline





1) Adversarial Attacks on Neural Network Policies
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

2) Emergence of Grounded Compositional Language in Multi-Agent Populations
Igor Mordatch, Pieter Abbeel

3) Autonomous Helicopter Flight
Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng

Challenges in Helicopter Control

- Unstable
- Nonlinear
- Complicated dynamics
  - Air flow
  - Coupling
  - Blade dynamics
- Noisy estimates of position, orientation, velocity, angular rate (and perhaps blade and engine speed)

Success Stories: Hover and Forward Flight

- Just a few examples:
  - Bagnell & Schneider, 2001;
  - LaCivita, Papageorgiou, Messner & Kanade, 2002;
  - Ng, Kim, Jordan & Sastry 2004a (2001); Ng et al., 2004b;
  - Roberts, Corke & Buskey, 2003;
  - Saripalli, Montgomery & Sukhatme, 2003;
  - Shim, Chung, Kim & Sastry, 2003;
  - Doherty et al., 2004;
  - Gavrilets, Martinos, Mettler and Feron, 2002.
- Varying control techniques: inner/outer loop PID with hand or automatic tuning, H1, LQR, …

[Ng, Coates, Tse, et al, 2004]

Alan Szabo – Sunday at the Lake

One of our first attempts at autonomous flips
[using similar methods to what worked for ihover]

Target trajectory: meticulously hand-engineered
Model: from (commonly used) frequency sweeps data

Stationary vs. Aggressive Flight

- Hover / stationary flight regimes:
  - Restrict attention to specific flight regime
  - Extensive data collection = collect control inputs, position, orientation, velocity, angular rate
  - Build model + model-based controller

→ Successful autonomous flight.

- Aggressive flight maneuvers --- additional challenges:
  - **Task description**: What is the target trajectory?
  - **Dynamics model**: How to obtain accurate model?

Aggressive, Non-Stationary Regimes

- Gavrilets, Martinos, Mettler and Feron, 2002
  - 3 maneuvers: split-S, snap axial roll, stall-turn
  - Key: Expert engineering of controllers after human pilot demonstrations

Sunday in Open Loop

Aggressive, Non-Stationary Regimes

- Our work:
  - Key: Learn controllers from human pilot demonstrations + RL
  - Wide range of aggressive maneuvers
  - Maneuvers in rapid succession

Learning Dynamic Maneuvers

- **Learning a target trajectory**
- Learning a dynamics model
- Autonomous flight results

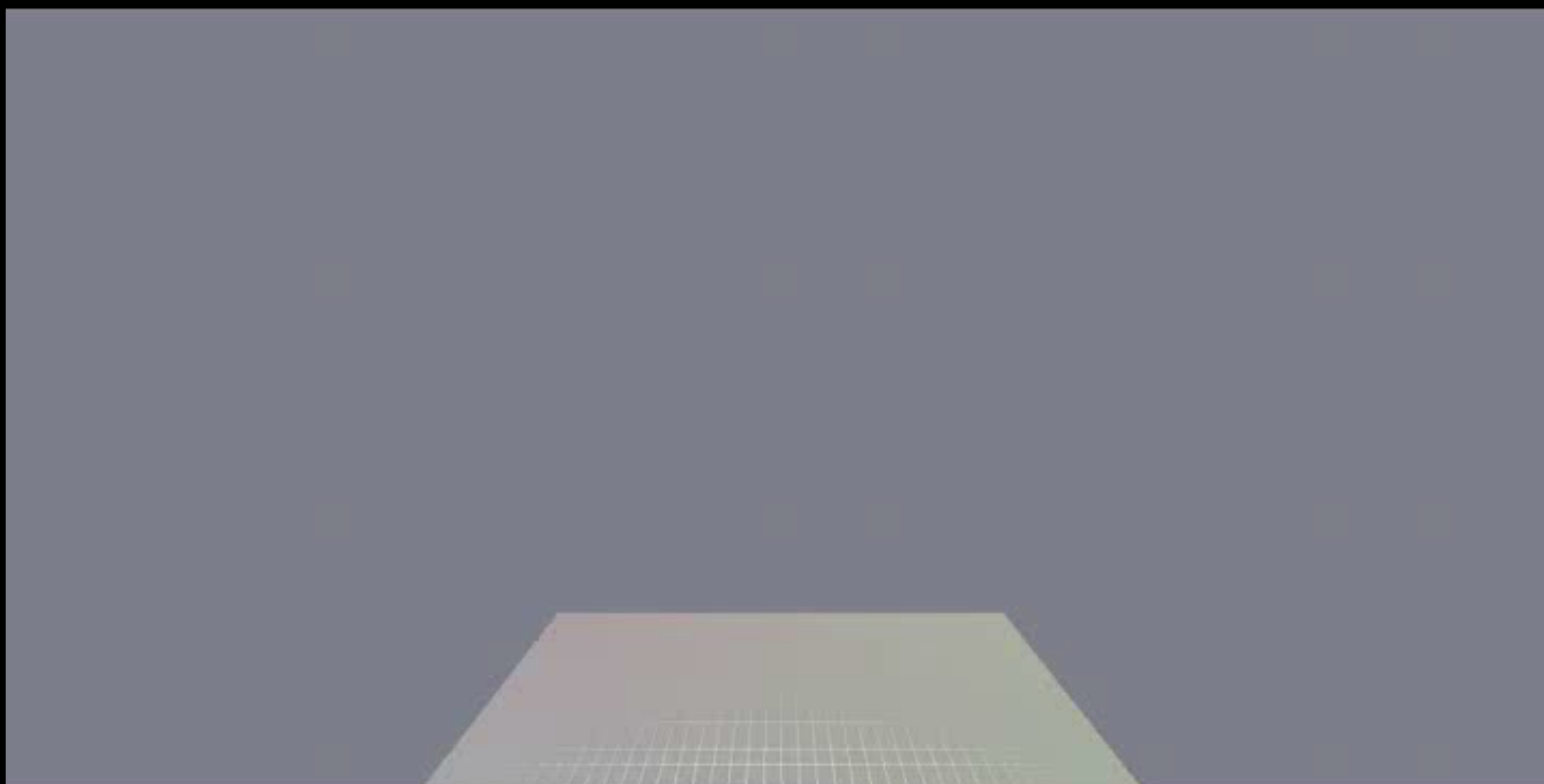
Target Trajectory

- Difficult to specify by hand:
  - Required format: position + orientation over time
  - Needs to satisfy helicopter dynamics

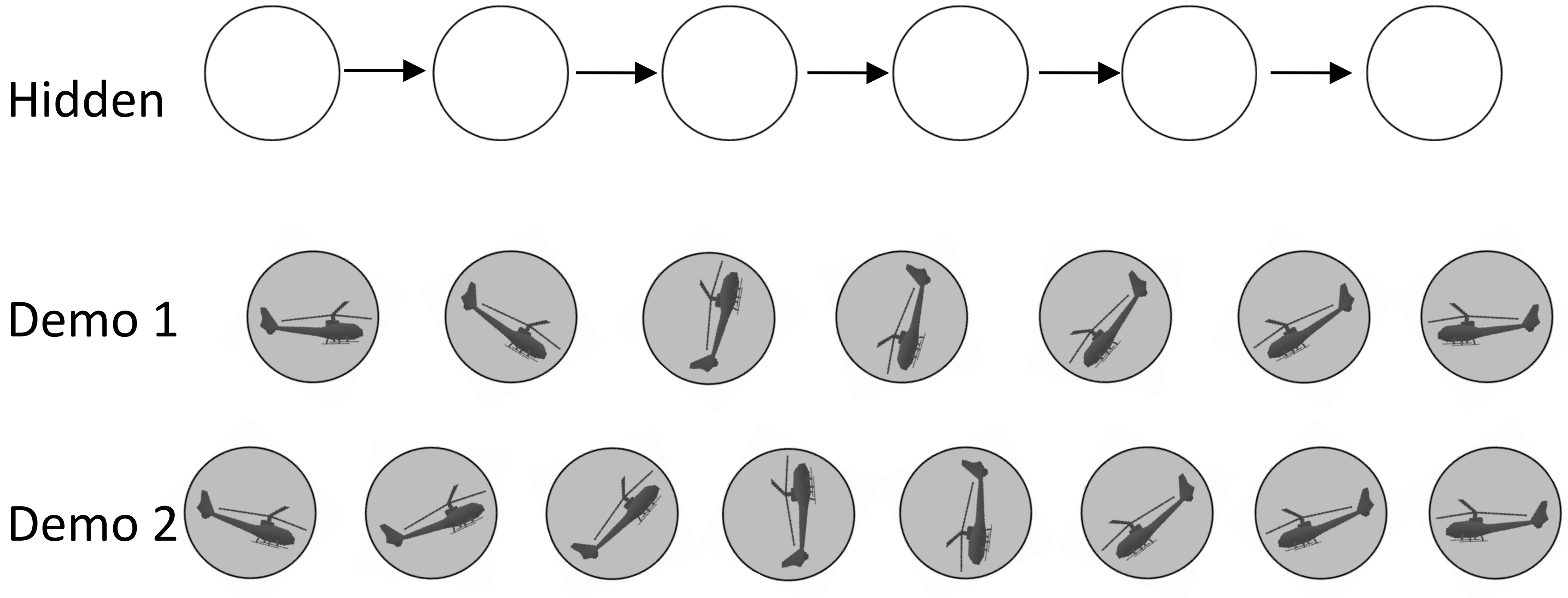
- Our solution:
  - Collect demonstrations of desired maneuvers
  - Challenge: extract a clean target trajectory from many suboptimal/ noisy demonstrations

Abbeel, Coates, Ng, IJRR 2010

Expert Demonstrations



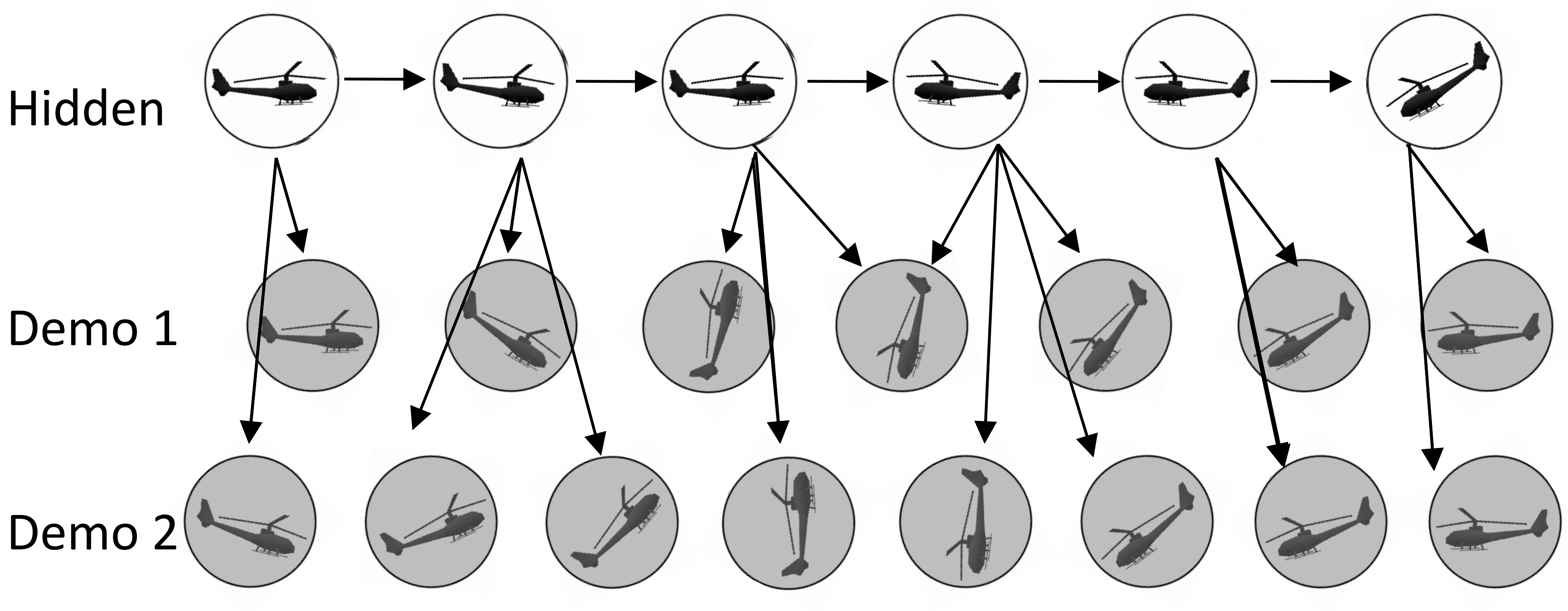
Learning a Trajectory



- HMM-like generative model
  - Dynamics model used as HMM transition model
  - Demos are observations of hidden trajectory
- Problem: how do we align observations to hidden trajectory?

Abbeel, Coates, Ng, IJRR 2010

Learning a Trajectory



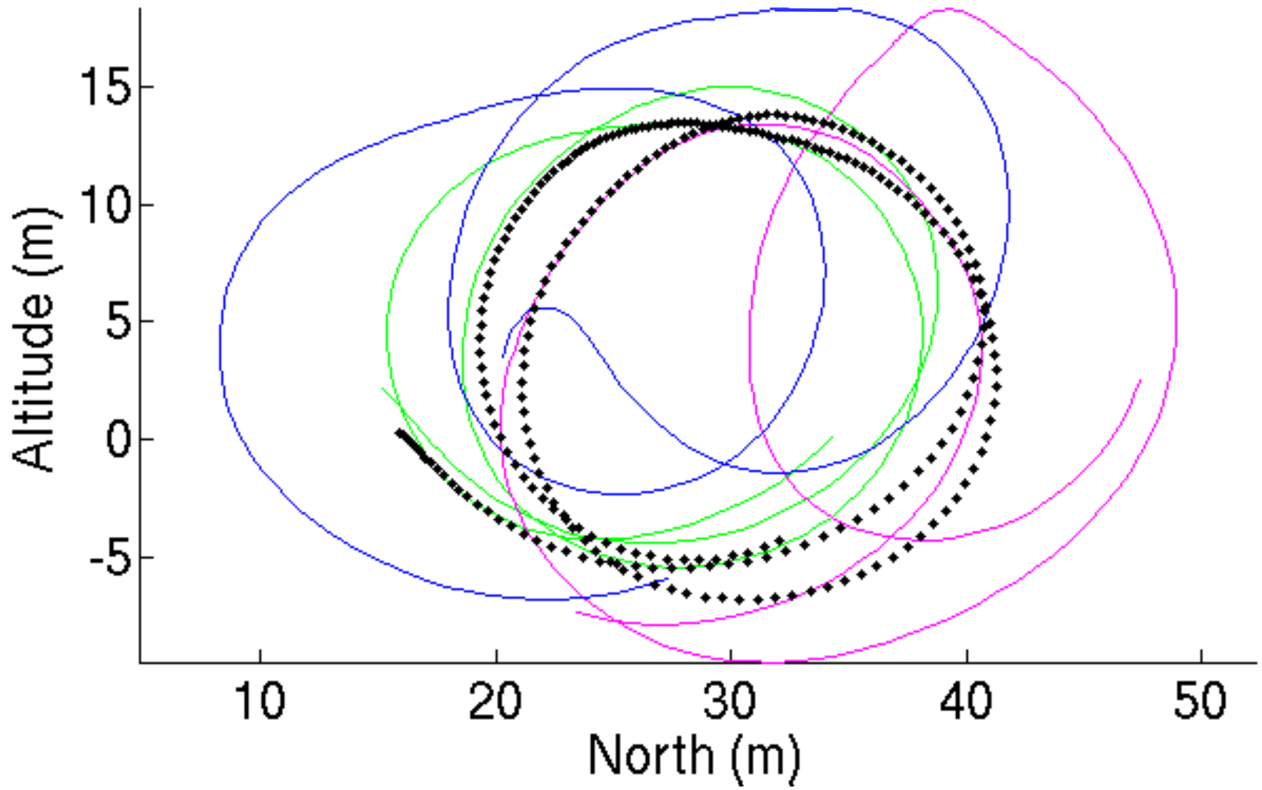
- Dynamic Time Warping (Needleman&Wunsch 1970 Sakoe&Chiba, 1978)
- Extended Kalman filter / smoother

Abbeel, Coates, Ng, IJRR 2010

Results: Time-Aligned Demonstrations

- White helicopter is inferred "intended" trajectory.

Results: Loops



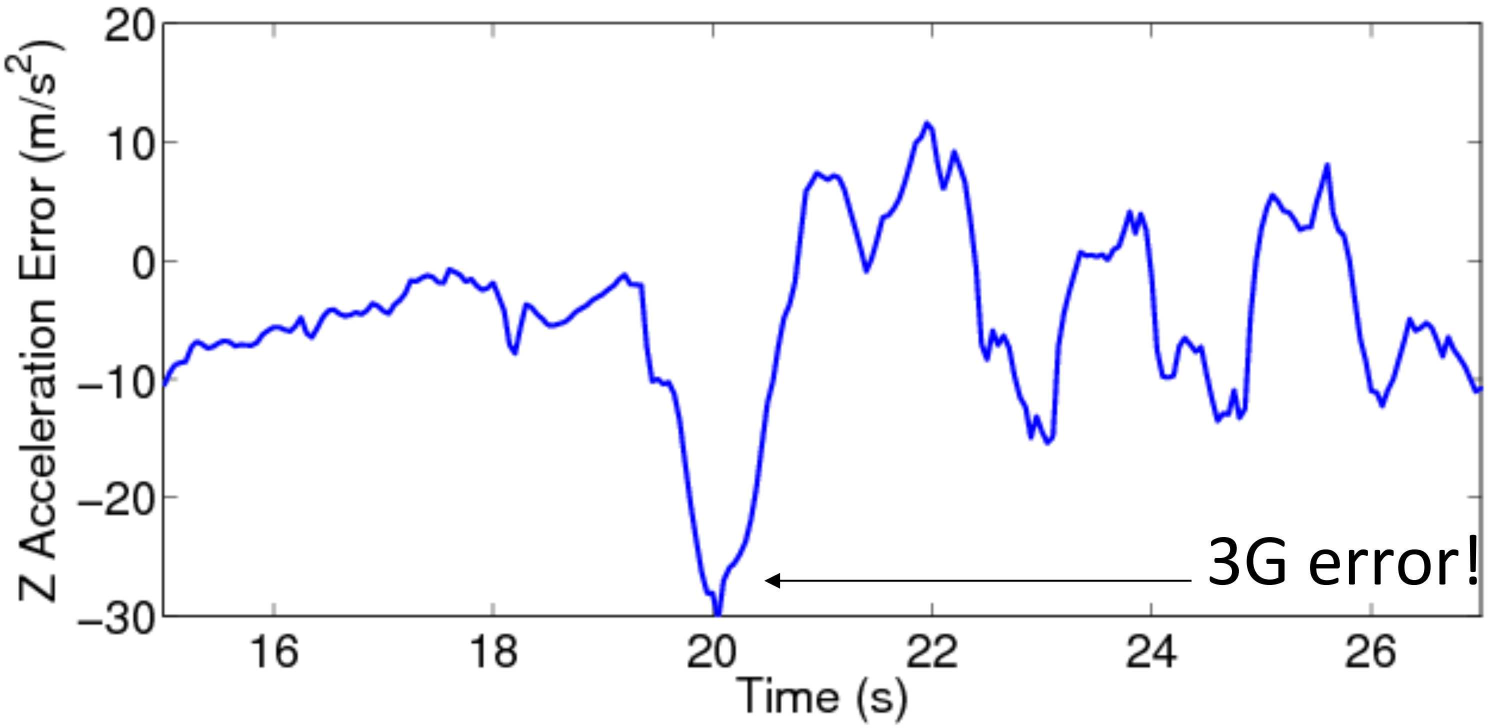
Even without prior knowledge, the inferred trajectory is much closer to an ideal loop.

Abbeel, Coates, Ng, IJRR 2010

Learning Dynamic Maneuvers

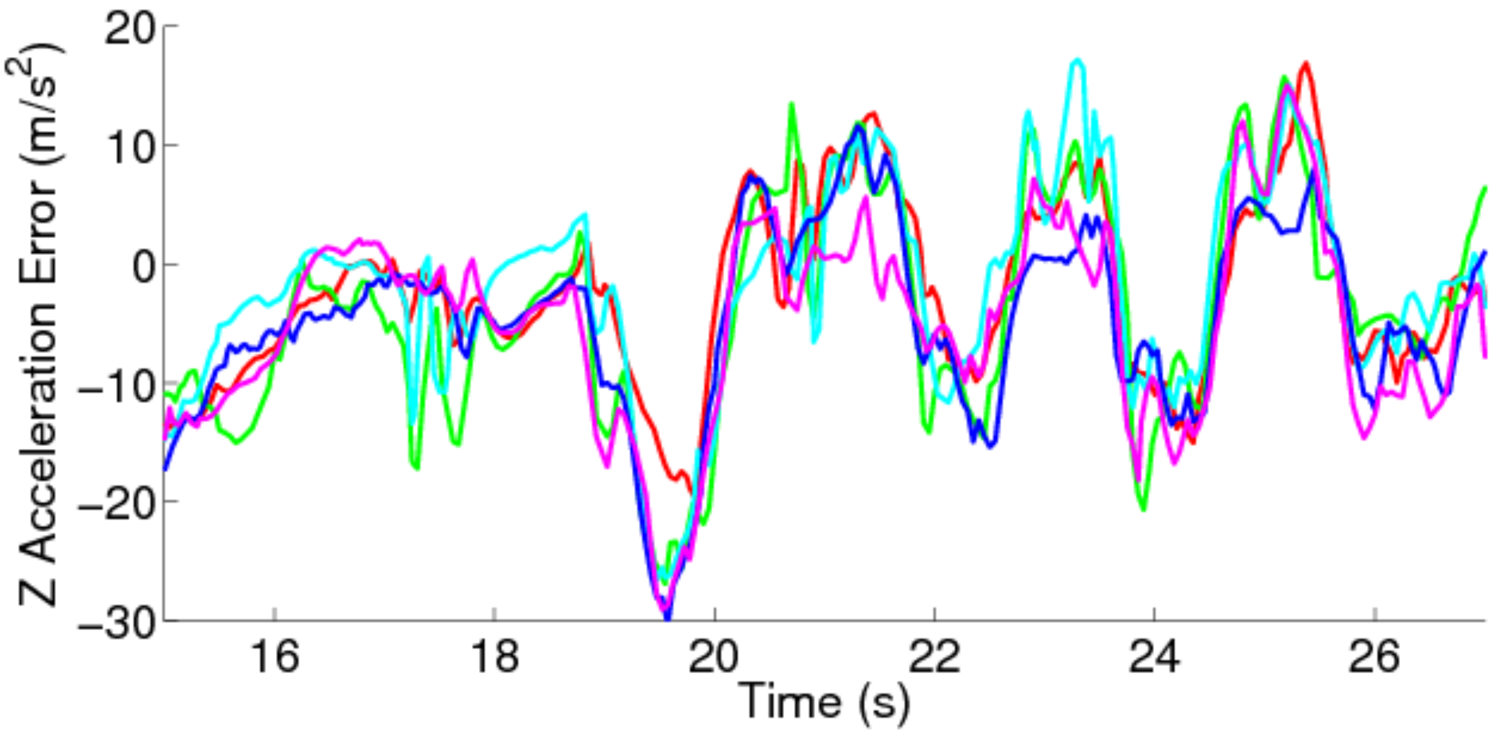
- Learning a target trajectory
- **Learning a dynamics model**
- Autonomous flight results

Standard Modeling Approach



Abbeel, Coates, Ng, IJRR 2010

Key Observation



Errors observed in the "baseline" model are clearly consistent after aligning demonstrations.

Abbeel, Coates, Ng, IJRR 2010

Key Observation

- If we fly the same trajectory repeatedly, errors are consistent over time once we align the data.
  - There are many unmodeled variables that we can't expect our model to capture accurately.
    - Air (!), actuator delays, etc.
  - If we fly the same trajectory repeatedly, the hidden variables tend to be the same each time.

~ muscle memory for human pilots

Abbeel, Coates, Ng, IJRR 2010

Trajectory-Specific Local Models

- Learn locally-weighted model from aligned demonstrations
  - Since data is aligned in time, we can weight by *time* to exploit repeatability of unmodeled variables.
  - For model at time t: $$W(t') = e^{-\frac{(t-t')^2}{\sigma^2}}$$
  - Obtain a model for each time t into the maneuver by running weighted regression for each time t

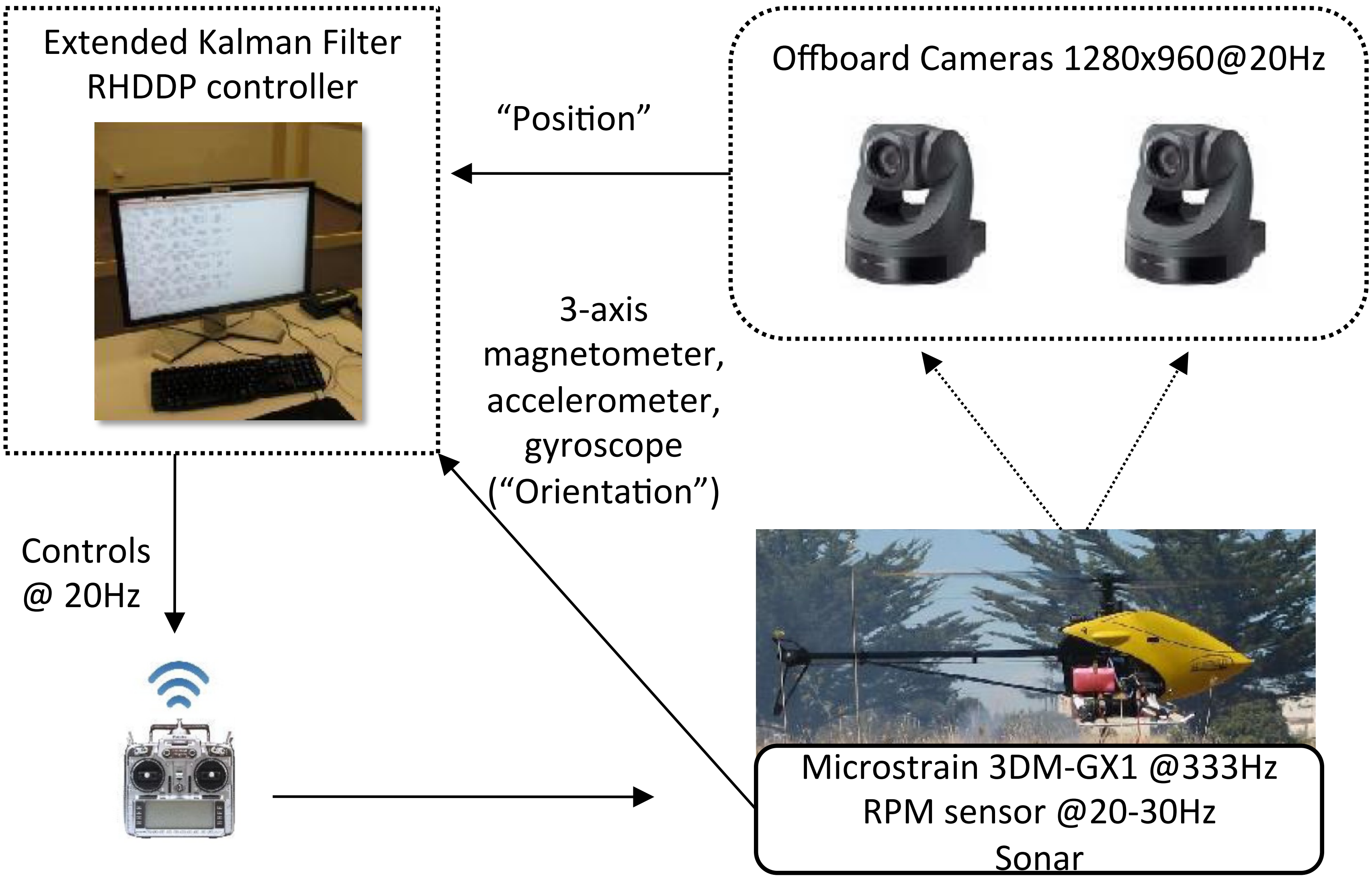
Abbeel, Coates, Ng, IJRR 2010

Learning Dynamic Maneuvers

- Learning a target trajectory
- Learning a dynamics model
- **Autonomous flight results**

Abbeel, Coates, Ng, IJRR 2010

Experimental Setup

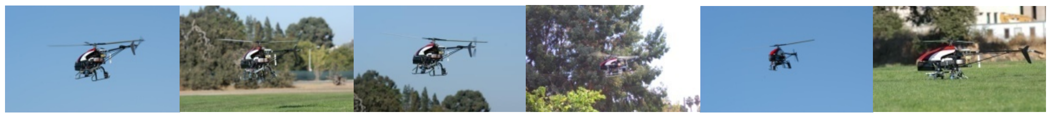


Abbeel, Coates, Quigley, Ng, NIPS 2007

Experimental Procedure

1. Collect sweeps to build a baseline dynamics model

2. Our expert pilot demonstrates the airshow several times.

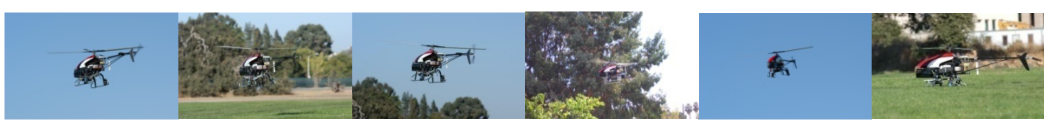


3. Learn a target trajectory.

4. Learn a dynamics model.

5. Find the optimal control policy for learned target and dynamics model.

6. Autonomously fly the airshow



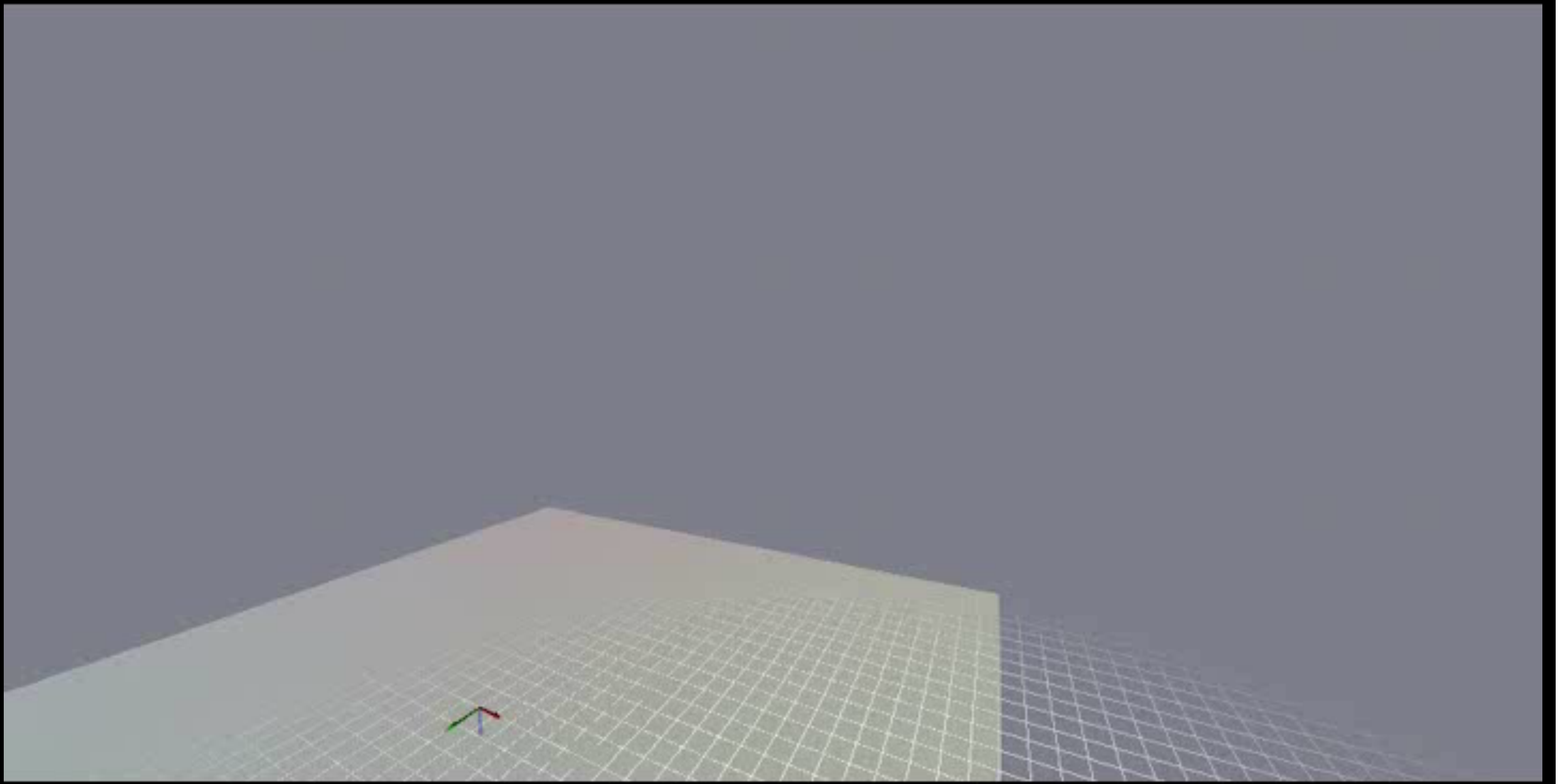
7. Learn an improved dynamics model. Go back to step 4.

→ **Learn to fly new maneuvers in < 1hour.**

Abbeel, Coates, Ng, IJRR 2010

Results: Autonomous Airshow

Results: Flight Accuracy

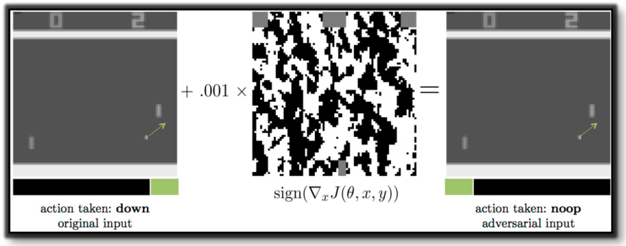


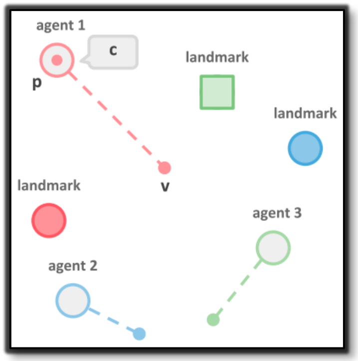
Autonomous Autorotation Flights

Abbeel, Coates, Hunter, Ng, ISER 2008

Chaos ["flip/roll" parameterized by yaw rate]

Summary





1) ***Adversarial Attacks on Neural Network Policies***

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

2) ***Emergence of Grounded Compositional Language in Multi-Agent Populations***

Igor Mordatch, Pieter Abbeel

3) ***Autonomous Helicopter Flight***

Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng

Current / Future Directions

- Faster learning / Hierarchy
  - Exploration (Stadie, Levine, Abbeel 2015; Houthooft, Duan, Chen, Schulman Abbeel, 2016)
  - Meta-learning: RL2 (Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016); MAML (Finn, Abbeel, Levine, 2017)
- Transfer learning
  - Modular networks (Devin, Gupta, Darrell, Abbeel, Levine, 2017) ; Invariant feature spaces (Gupta Devin, Liu, Abbeel, Levine, 2017)
  - Domain randomization (Tobin, Fong, Schneider, Zaremba, Abbeel, 2017)
- Safe learning
  - Kahn, Villaflor, Pong, Abbeel, Levine, 2017; Held, McCarthy, Zhang, Shentu, Abbeel, 2016
- Unsupervised / Semisupervised learning
  - InfoGAN (Chen, Duan, Houthooft, Schulman, Sutskever, Abbeel 2016), VLAE (Chen, Kigma, Salimans, Duan, Dhariwal, Schulman, Sutskever, Abbeel, 2017)
  - Semisupervised RL (Finn, Yu, Fu, Abbeel, Levine, 2017)
- Grounded language / Multi-agent
  - "Inventing" language (Mordatch & Abbeel, 2017)
- Imitation
  - First-person from VR Tele-op (McCarthy, Zhang, Jow, Lee, Goldberg, Abbeel, 2017)
  - Third-person (Stadie, Abbeel, Sutskever, 2017)
- Value alignment / AI Safety
  - CIRL (Hadfield-Menell, Dragan, Abbeel, Russell, 2016), Off-switch (Hadfield Menell, Dragan, Abbeel, Russell, 2017)
  - Communication (Huang, Held, Abbeel, Dragan, 2017)