Advanced Topics in Imitation Learning & Safety

3/1/2017

Chelsea Finn

Advanced Topics in Imitation & Safety

1. Imitating humans: handling domain shift
2. Safety while learning
3. Improving imitation learning from experts

Note: This is an open area of research.
[Next time, there will be more theory.]

Advanced Topics in Imitation & Safety

1. **Imitating humans: handling domain shift**
2. Safety while learning
3. Improving imitation learning from experts

Imitation learning beyond drones and cars

Q: How do you provide demonstrations for an robotic arm or humanoid?

Kinesthetic Teaching

Muelling et al. '13

Teleoperation

Pastor et al. '12

VR Teleoperation

Zoe McCarthy '16

Imitation learning beyond drones and cars

Q: How do you provide demonstrations for an robotic arm or humanoid?

A: Most popularly:
- kinesthetic teaching
- teleoperation

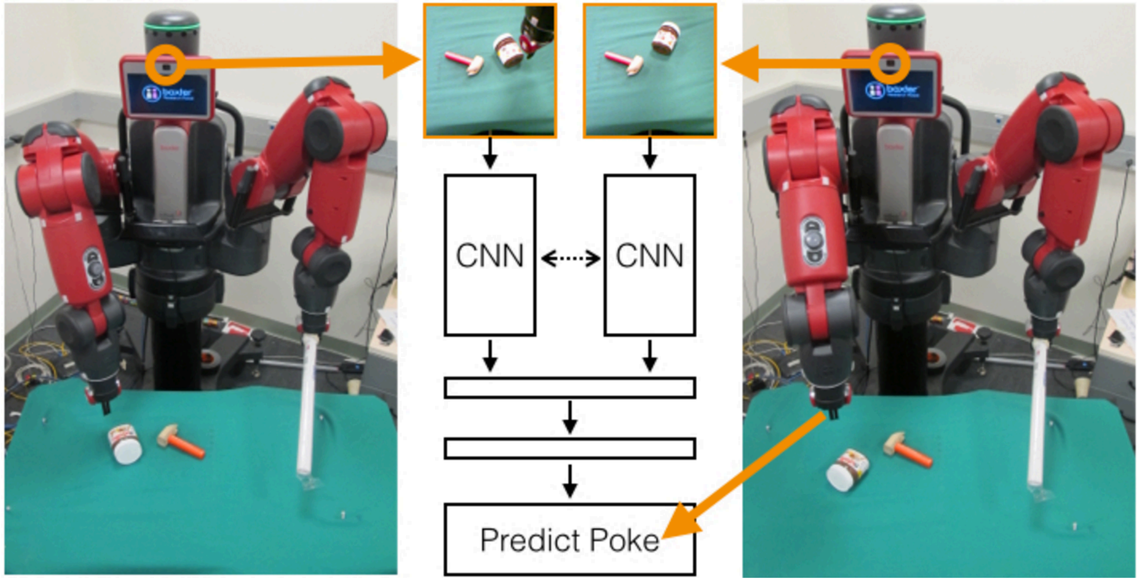
This lecture: beyond kinesthetic teaching & teleoperation

Why is domain shift a problem?

- Humans can do things that robots can’t do (and vice versa)
- Humans look different than robots

With expressive function approximation, domain shift becomes more of an issue.
[With linear functions, we will just underfit.]

Last time: models with images

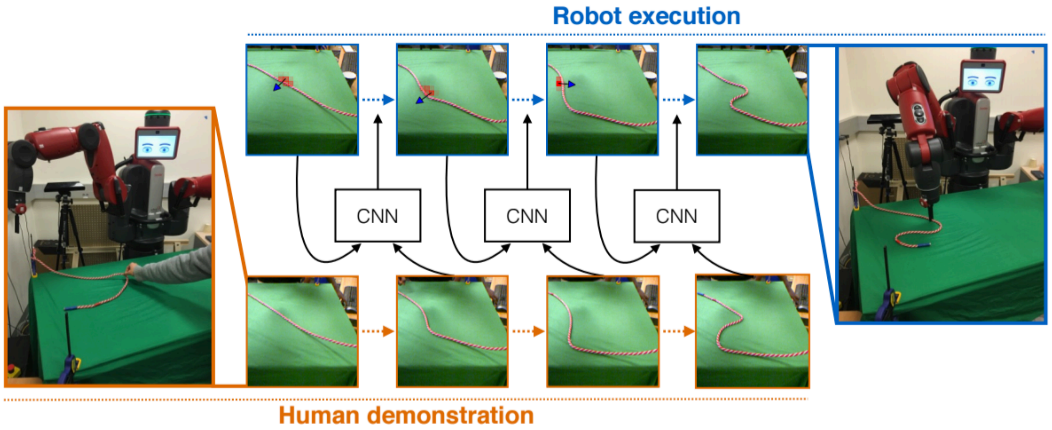


Limitation: can't plan with inverse model

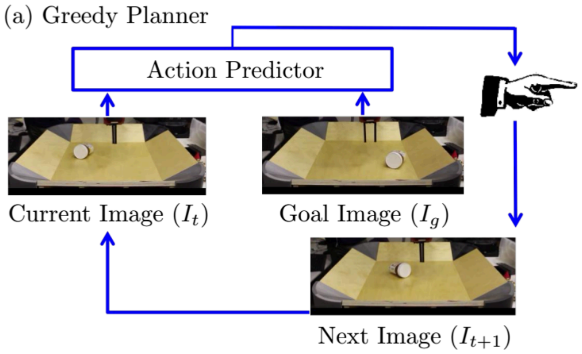
Case Study: Imitating Intermediate Goals

Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation

Ashvin Nair* Dian Chen* Pulkit Agrawal*
Phillip Isola Pieter Abbeel Jitendra Malik Sergey Levine

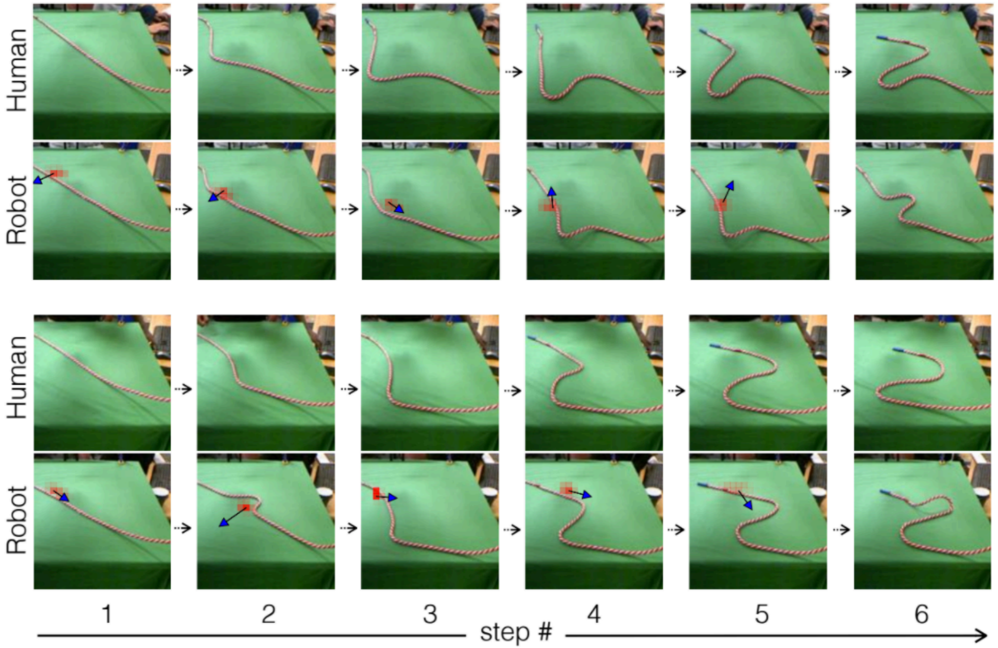


Imitation of high-level goals



Can we get high-level subgoals from humans?

Handling domain shift



- Observation: Human is removed from scene. (not always a full solution)
- More importantly, human provides *high-level* goals

Another reference: Modular Multi-task Reinforcement Learning with Policy Sketches, Andreas et al ‘16

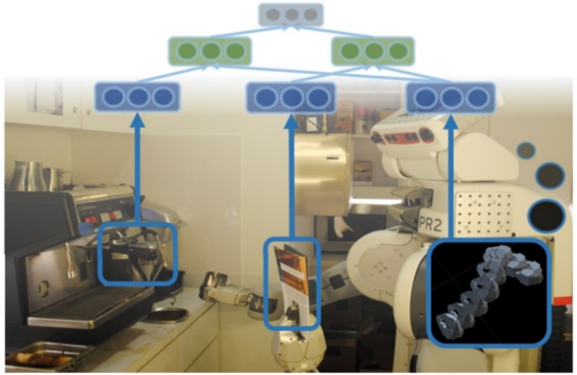
Limitations

- Inverse model may ignore important parts of the image
  - Only needs to model enough to determine the action
- Requires images of goal for each step
  - May be hard to obtain for some tasks (language?)

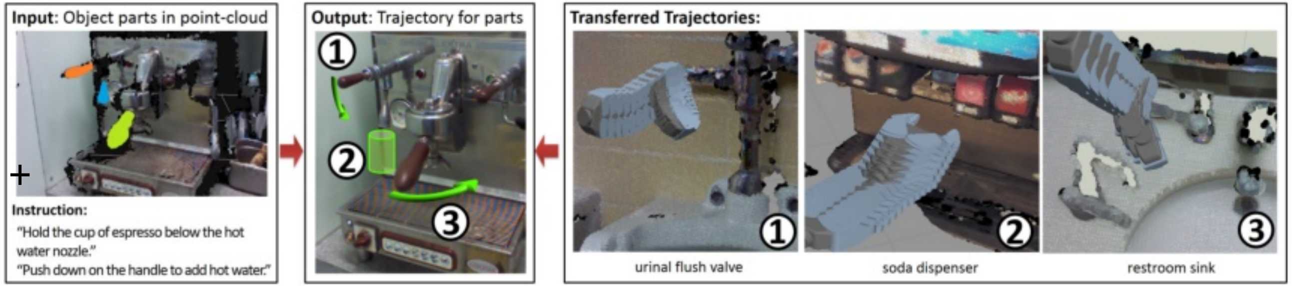
Case Study: Multimodal Imitation

Robobarista: Object Part based Transfer of Manipulation Trajectories from Crowd-sourcing in 3D Pointclouds

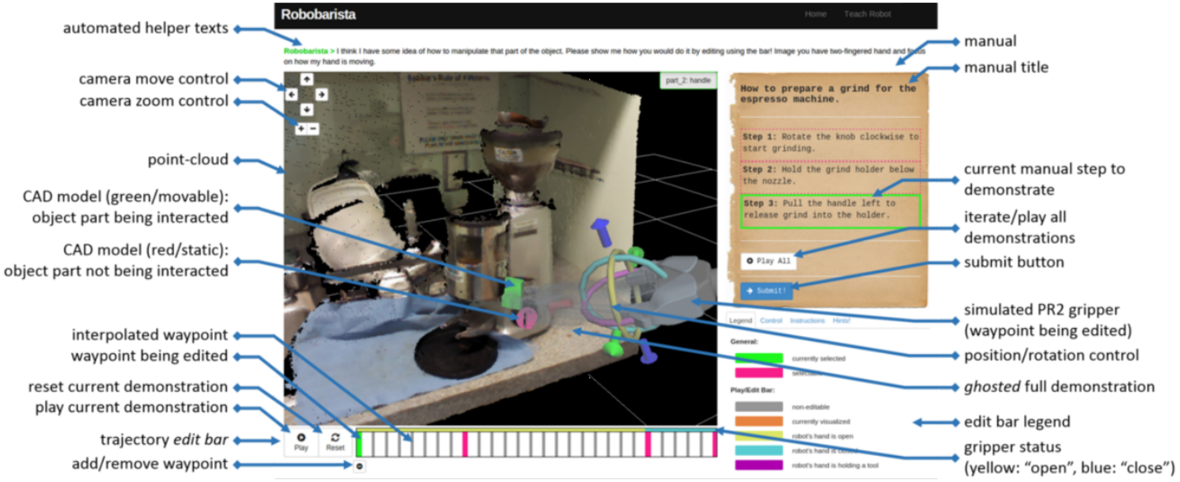
Jaeyong Sung, Seok Hyun Jin, and Ashutosh Saxena



Goal:
convert
text instruction +
point cloud
into robot
trajectory

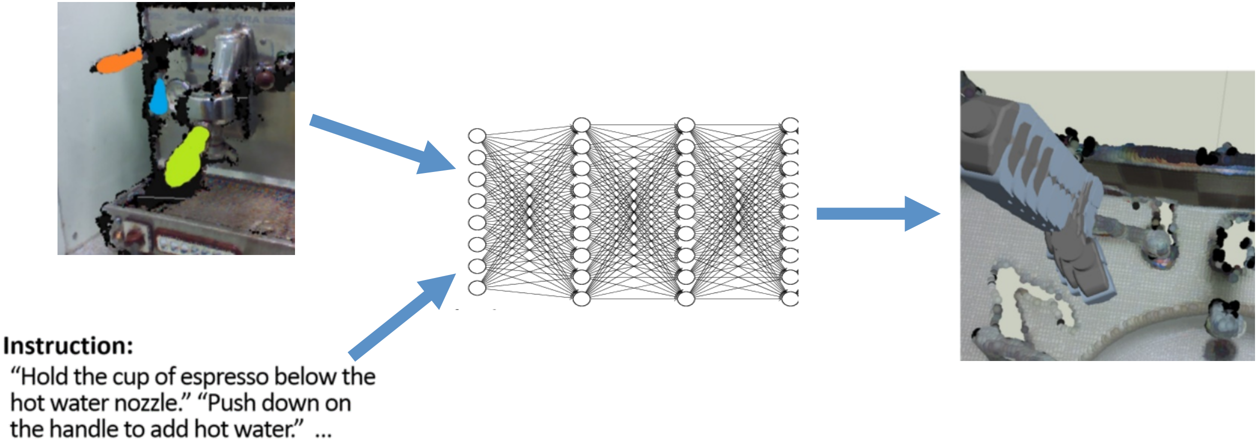


Demonstration Interface:



How to train the model?

"policy" model:

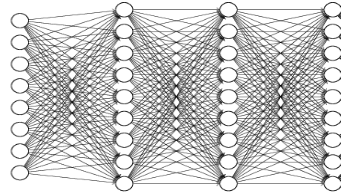


- Non-unique output
- Policy must learn to output good, detailed trajectories
  - Not as hard as generating realistic images
  - But still hard – lots of **precision** required!

How to train the model?

"critic" model:

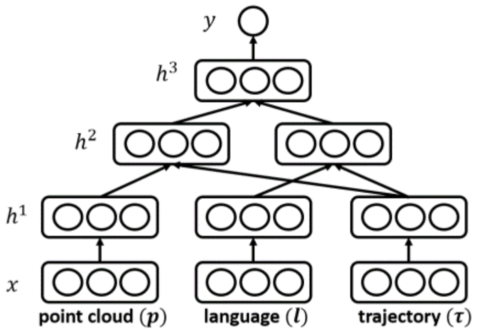
Instruction:
"Hold the cup of espresso below the hot water nozzle." "Push down on the handle to add hot water." ...



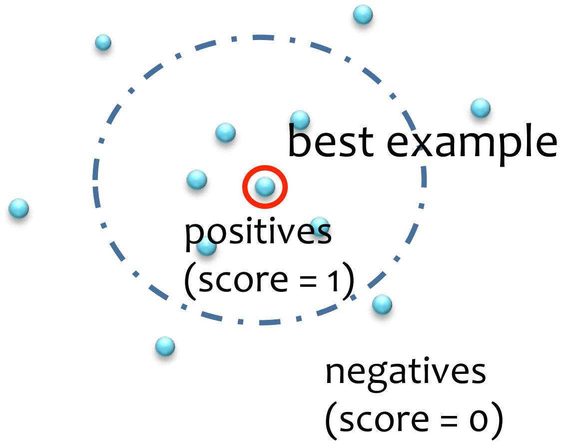
score

- Can assign good score to multiple trajectories
- Search for trajectories (e.g. nearest neighbor queries)
  - Trajectories are object-relative, so nearest neighbor is OK

How to train?



- Pretrain each encoder as autoencoder
- Assign a score to each demonstration trajectory
  - Trajectory that is the most similar to other trajectories is "best" (most representative)
  - Inner 50% are "good"
  - Outer 50% are "bad"
  - Train score function for binary classification




Input :
point cloud :
language : "hold below the nozzle"

Limitations

- Requires object-centric trajectories
  - Assumes nearest-neighbor queries are a reasonable way to get trajectories
  - Must have pre-segmented object parts (e.g. handles)
- Requires large number of demonstrations (1225)
  - Enough to determine positives and negatives
  - Multiple demonstrations per object and per text command

Can we combine with RL to require fewer demonstrations?

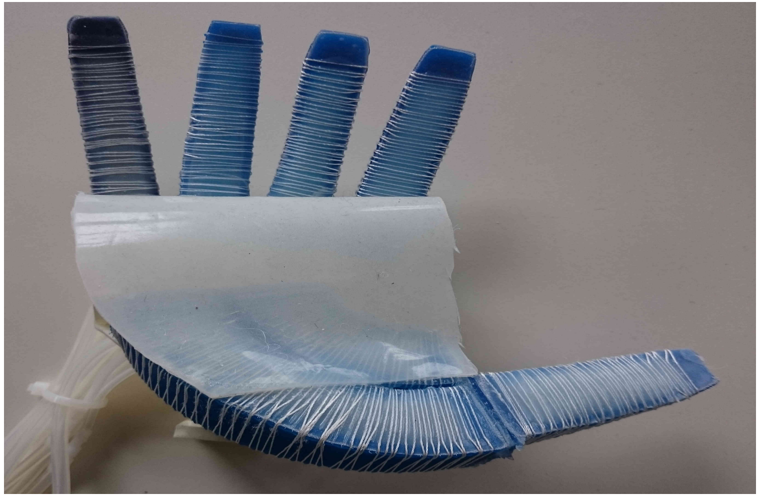
Case Study: Object-Centric Demonstrations

Learning Dexterous Manipulation for a Soft Robotic Hand from Human Demonstrations

Abhishek Gupta[1] Clemens Eppner[2] Sergey Levine[1] Pieter Abbeel[1]

RBO Hand 2

- Soft, Cheap, Compliant
- Inflate/deflate chambers using pneumatic actuators.
- 7 DoF – 1 on each finger, 3 on thumb, w/ pressure sensors for each chamber



Slide from Abhishek Gupta

Challenges w/ RBO Hand 2

- Cannot use traditional control methods:
  - Poor position and pressure sensing
  - Noisy actuation

- Difficult to teleoperate
  - **Kinesthetic teaching infeasible**
  - Data glove hard to use
  - RBO Hand 2 is not quite anthropomorphic

Slide from Abhishek Gupta

Learning from Demonstrations

- Use demonstrations to help acquire dexterous skills

- Use **“<u>object-centric</u>”** demonstrations.
  – Only care about motion of manipulated objects

  – Can be given by a human using their own hands

Slide from Abhishek Gupta

Algorithm motivation

- With GPS, train controllers to imitate object-centric demonstrations.
- Train neural network policy to generalize over individual controllers.
- Leaves the questions:
  - **Which demonstration/combination of demonstrations can each controller imitate most closely?**
  - **How should the controller imitate the demonstrations?**

Slide from Abhishek Gupta

Algorithm overview

- With GPS, train controllers to imitate object-centric demonstrations.
- Train neural network policy to generalize over individual controllers.

Slide from Abhishek Gupta

Problem Definition

- Model demonstrations as a mixture of Gaussians

$$d(\tau) = \sum_{i}^{D} v_i d_i(\tau) = \sum_{i}^{D} \sum_{j}^{C} b_{ij} d_i(\tau)$$

- Model controllers as a mixture of Gaussians

$$p(\tau) = \sum_{j}^{C} w_j p_j(\tau) = \sum_{j}^{C} \sum_{i}^{D} a_{ij} p_j(\tau)$$

- Objective is to minimize divergence between these distributions

$$\min_{p} D_{KL}(p(\tau)||d(\tau))$$

Slide from Abhishek Gupta

Algorithm Derivation

- Instead, we use a variational upper bound, using Jensen's inequality.

$$D_{KL}(p(\tau)||d(\tau)) \leq \sum_{i,j} a_{ij} D_{KL}\left(p_j(\bar{\tau})||d_i(\bar{\tau})\right) + D_{KL}\left(a||b\right)$$

- Minimizing upper bound, optimization problem becomes

$$\min_{p,a,b} \sum_{i,j} a_{ij} D_{KL}\left(p_j(\bar{\tau})||d_i(\bar{\tau})\right) + D_{KL}\left(a||b\right)$$

Slide from Abhishek Gupta

Solving Optimization

- We can perform coordinate descent wrt {a,b} and p to get 2 phases:
  - Correspondence weight learning (a,b)
    - Easy to find closed form solutions – convex in a,b
  - Controller optimization (p)

Slide from Abhishek Gupta

Controller Optimization

– The optimization w.r.t. p uses the fixed correspondence weights a_{ij} and minimizes weighted l_2 distance between controllers and demonstrations.

$$\min_{p_j(\tau)} \sum_{t,i} \frac{a_{ijt}}{\sum_{i'} a_{i'jt}} E_{\bar{x_t} \sim p_j(\bar{\tau})} \left[\frac{1}{2} (\bar{x_t} - \mu_{it})^T \Sigma_i^{-1} (\bar{x_t} - \mu_{it}) \right] - H(p_j(\bar{\tau}))$$

Slide from Abhishek Gupta

Algorithm Overview

Collect samples on RBO Hand 2

Fit dynamics using samples

Estimate Correspondence Weights

Perform optimal control to minimize weighted sum of distances to demonstrations

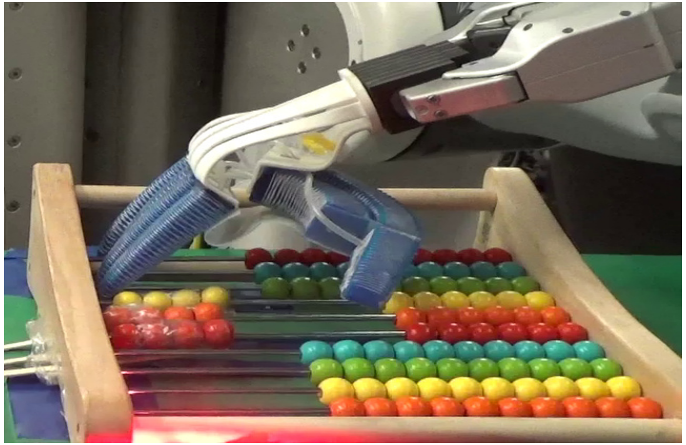
BADMM

Perform supervised learning to learn a global NN policy

Slide from Abhishek Gupta

Experiments

- Evaluated algorithm on 3 different real world tasks using the RBO Hand 2
  - Valve rotation
  - Pushing beads of abacus
  - Bottle grasping



Slide from Abhishek Gupta

Baselines

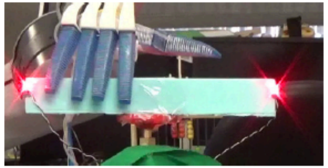
- **Hand designed baseline:** Controller with a hand-designed open loop policy
- **Single demo baseline:** A single controller trained to imitate a single demonstration.
- **Oracle:** Manually hard-assign single demonstrations to controllers

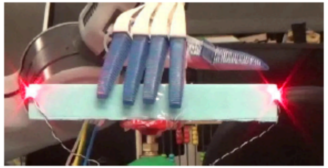
Slide from Abhishek Gupta

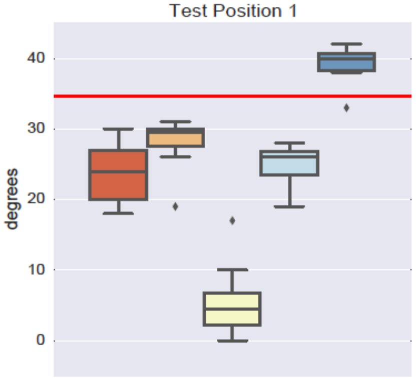
Valve Rotation

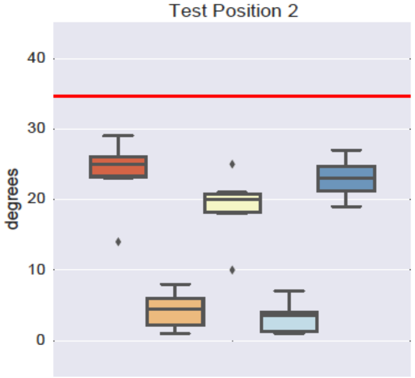
Slide from Abhishek Gupta

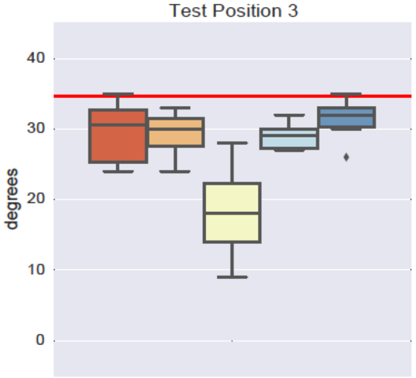
Results









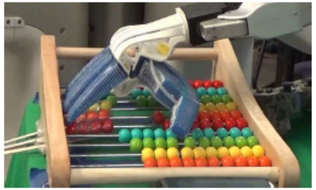


Slide from Abhishek Gupta

Pushing abacus beads

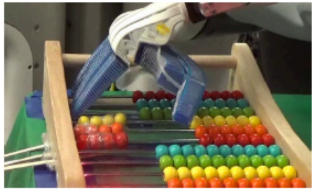
Slide from Abhishek Gupta

Results



Bead	Target	**Ours**	SingleDemo1	SingleDemo2	Oracle	HandDesign1	HandDesign2
1	8.4	**7.49 ± 0.47**	7.02 ± 0.50	6.33 ± 2.15	7.66 ± 0.23	8.38 ± 0.04	0 ± 0
2	0	**0.14 ± 0.18**	0.60 ± 0.69	7.08 ± 1.04	0.27 ± 0.42	0 ± 0	6.5 ± 0
3	0	**0.89 ± 1.00**	0.28 ± 0.18	1.23 ± 2.20	1.08 ± 0.72	0 ± 0	8.43 ± 0.29

Bead	Target	**Ours**	SingleDemo1	SingleDemo2	Oracle	HandDesign1	HandDesign2
1	8.4	**7.95 ± 0.19**	1.04 ± 2.15	7.27 ± 0.65	7.52 ± 0.66	0.00 ± 0.00	8.38 ± 0.08
2	0	**0.10 ± 0.10**	0.85 ± 1.21	0.19 ± 0.14	0.09 ± 0.11	0.00 ± 0.00	8.40 ± 0.00
3	0	**0.00 ± 0.00**	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00



Bead	Target	**Ours**	SingleDemo1	SingleDemo2	Oracle	HandDesign1	HandDesign2
1	8.4	**7.21 ± 0.69**	2.47 ± 2.22	3.39 ± 1.98	7.74 ± 0.23	0.00 ± 0.00	8.38 ± 0.05
2	0	**0.00 ± 0.00**	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
3	0	**0.00 ± 0.00**	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Slide from Abhishek Gupta

Bottle Grasping

- 10/10 successful grasps learned.

Slide from Abhishek Gupta

Benefits and Limitations

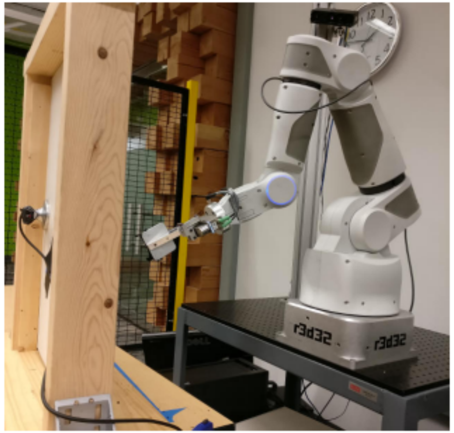
- Learn from a few demonstrations
  - Uses RL to learn how to how to mimic demonstration trajectory of object(s)
- Requires phase-space system for object pose
  - Gets object-centric demonstrations via ground truth pose
  - Less clear how raw pixels could be used

How can we get domain invariance with raw pixel observations?

Case Study: Using Domain-Invariant Features

UNSUPERVISED PERCEPTUAL REWARDS FOR IMITATION LEARNING

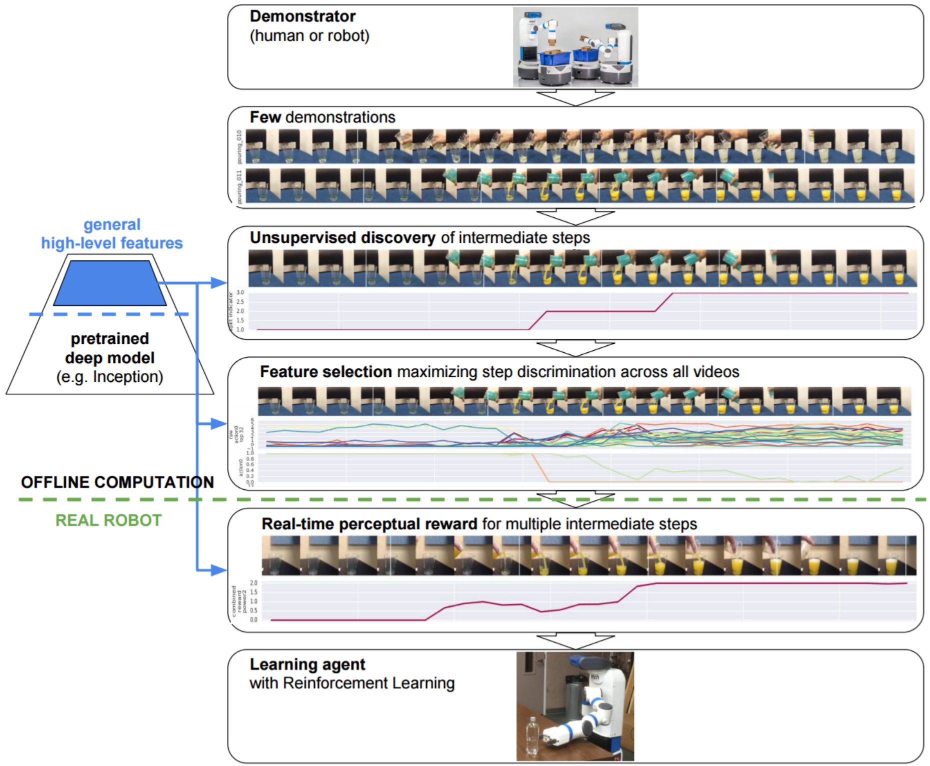
Pierre Sermanet, Kelvin Xu* & Sergey Levine
Google Brain

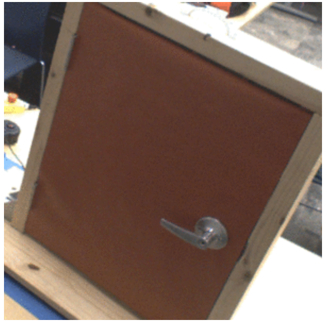


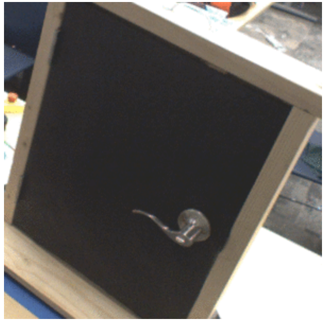
Main Idea: Leverage pre-trained image features

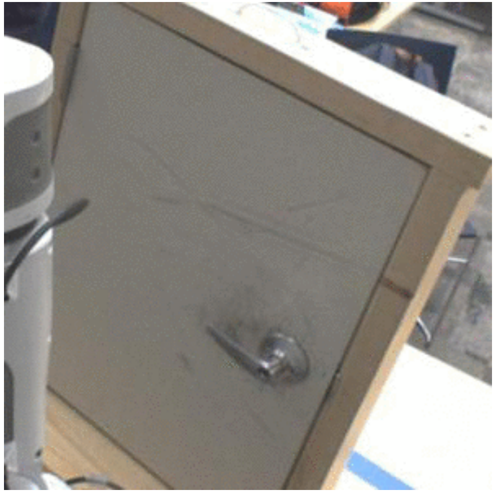
1. Collect demonstration videos & compute features on frames
2. Unsupervised discovery of N stages of the demonstration
3. Automatically select M most relevant features for each stage
4. Run RL to match features

Learning what Success Means









Sermanet, Xu, L. '16

Slide from Sergey Levine

How does this compare to using true reward?

Benefits and Limitations

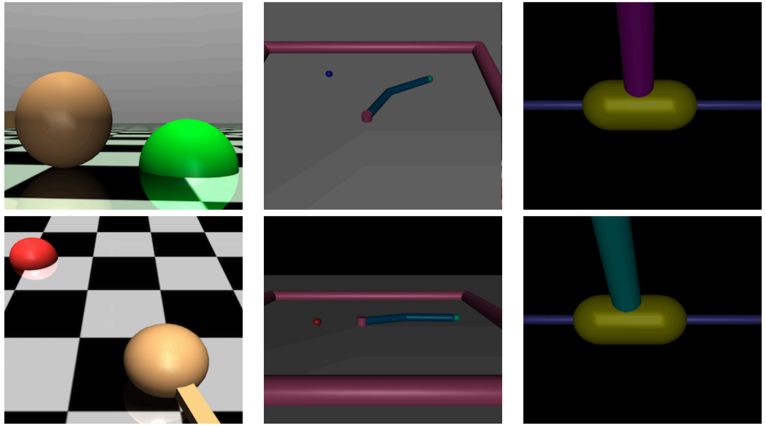
- Very simple and effective
- Learn from raw pixels
- Only as good as the features
- Only provides a success/failure classifier
  - Doesn't reason about outcomes or how the task can be solved
  - Agent can potentially fool the classifier

Can we reason about the task (i.e. the reward) using demonstrations?
Inverse RL - next lecture!

Note: Can optimize for domain invariance

Third Person Imitation Learning

Bradly C Stadie[1,2], **Pieter Abbeel**[1,3], **Ilya Sutskever**[1],
[1] OpenAI
[2] UC Berkeley, Department of Statistics
[3] UC Berkeley, Department of Electrical Engineering and Computer Science



Domain Shift – Conclusions

Ways to handle domain shift:

- Remove human from the scene
- Have humans provide high-level goals
- Object-centric demonstrations
- Use domain-invariant representations
- Optimize for domain-invariance

Advanced Topics in Imitation & Safety

1. Imitating humans: handling domain shift
2. **Safety while learning**
3. Improving imitation learning from experts

Recap: DAgger

DAgger: Dataset **A**ggregation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

how? just run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

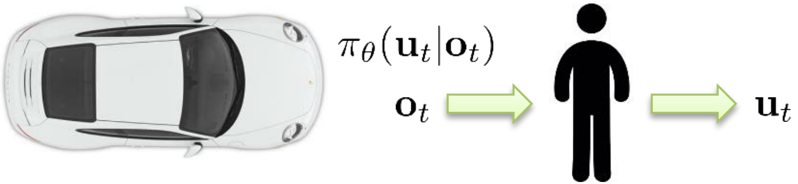
but need labels $\mathbf{u}_t$!

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \dots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{u}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

What's the problem?

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. $\boxed{\text{run } \pi_\theta(\mathbf{u}_t|\mathbf{o}_t)}$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. $\boxed{\text{Ask human to label } \mathcal{D}_\pi \text{ with actions } \mathbf{u}_t}$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



Ross et al. '11

A partial solution to both

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. $\boxed{\text{run } \pi_\theta(\mathbf{u}_t|\mathbf{o}_t)}$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. $\boxed{\text{Ask human to label } \mathcal{D}_\pi \text{ with actions } \mathbf{u}_t}$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Idea: Train a classifier to classify accuracy of π_θ for a given $\mathbf{o}_t$

1. Only request labels for observations where policy is inaccurate
2. In safety-critical applications: switch to safe, expert policy when accuracy below some threshold

Laskey et al. ICRA '16
Zhang & Oh arXiv '16

Ross et al. '11

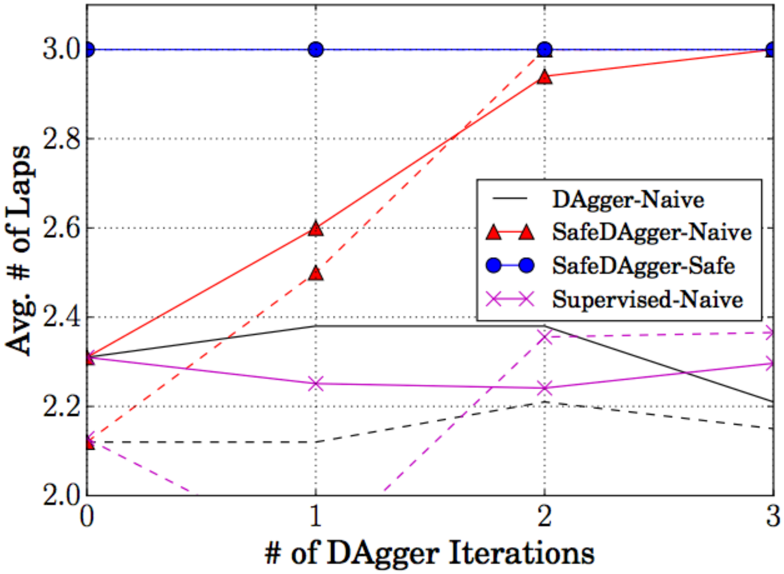
SafeDAgger (Zhang & Oh '16)

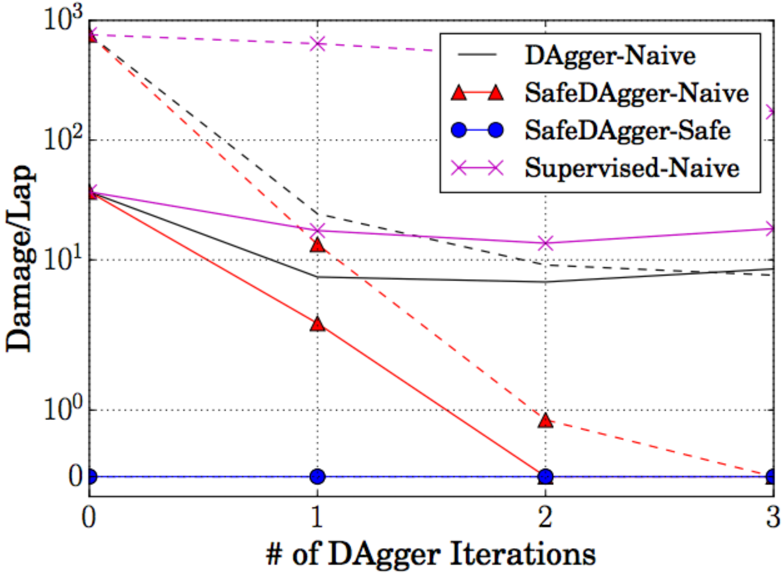
Algorithm 1 SafeDAgger Blue fonts are used to highlight the differences from the vanilla DAgger.

```
Collect D_0 using a reference policy π*
Collect D_safe using a reference policy π*
π_0 = arg min_π l_supervised(π, π*, D_0)
π_safe,0 = arg min_{π_safe} l_safe(π_safe, π_0, π*, D_safe ∪ D_0)
for i = 1 to M do
    Collect D' using the safety strategy using π_{i-1} and π_safe,i-1
    Subset Selection: D' ← {φ(s) ∈ D' | π_safe,i-1(π_{i-1}, φ(s)) = 0}
    D_i = D_{i-1} ∪ D'
    π_i = arg min_π l_supervised(π, π*, D_i)
    π_safe,i = arg min_{π_safe} l_safe(π_safe, π_i, π*, D_safe ∪ D_i)
end for
return π_M and π_safe,M
```

SafeDAgger (Zhang & Oh '16)

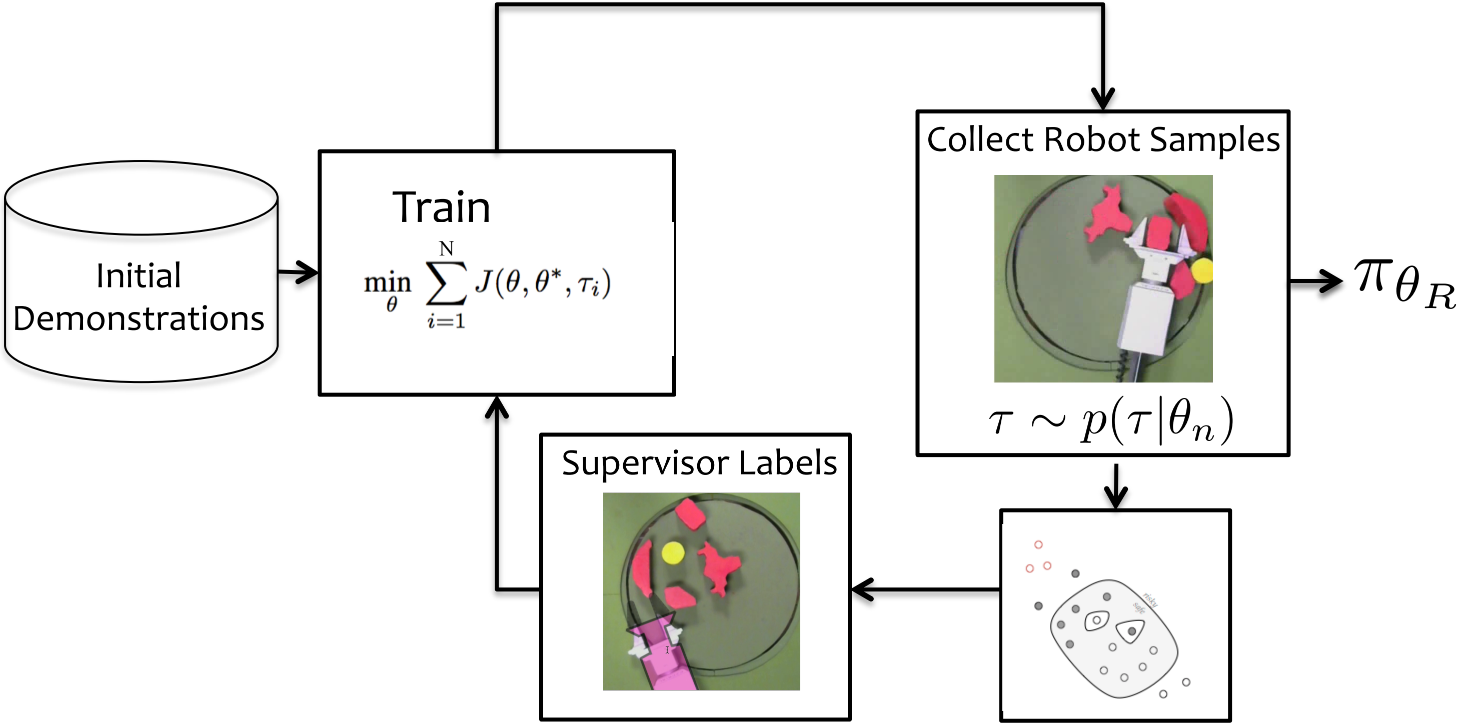
TORCS driving experiments



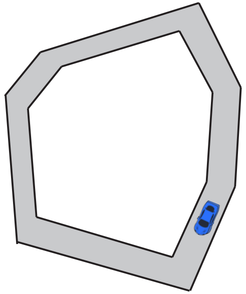


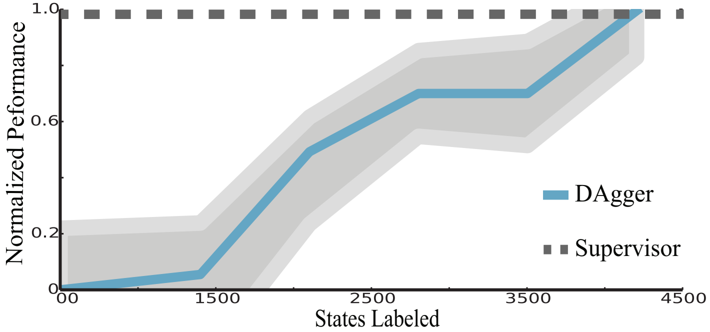
Solid line, no traffic; Dashed line, with traffic

SHIV: SVM-based Reduction in Human InterVention *(Laskey et al. '16)*

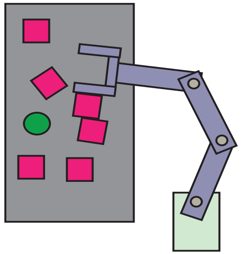


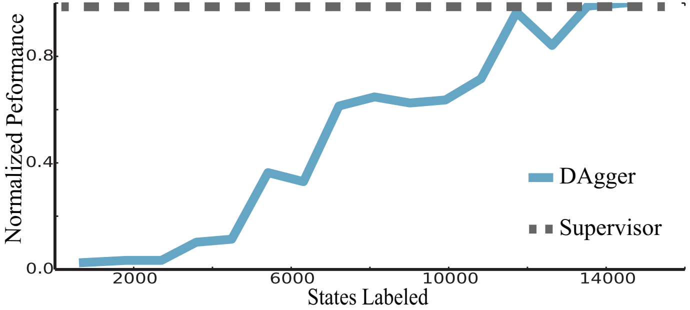
Slide adapted from Michael Laskey



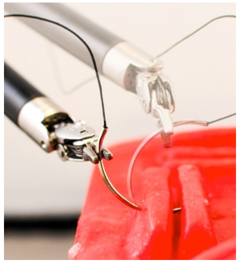


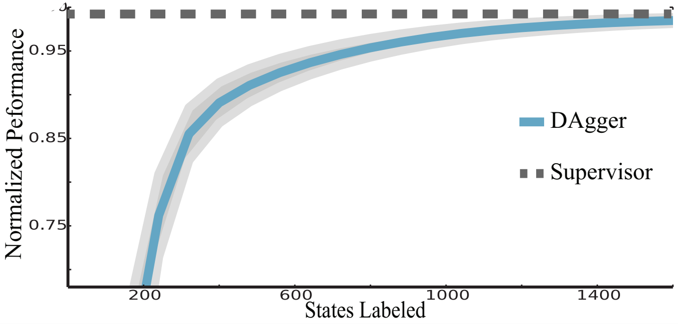
(a) Driving





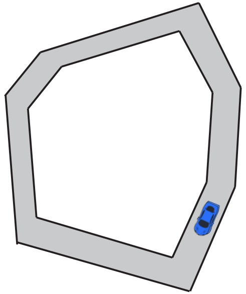
(b) Grasping in Clutter in Box2D

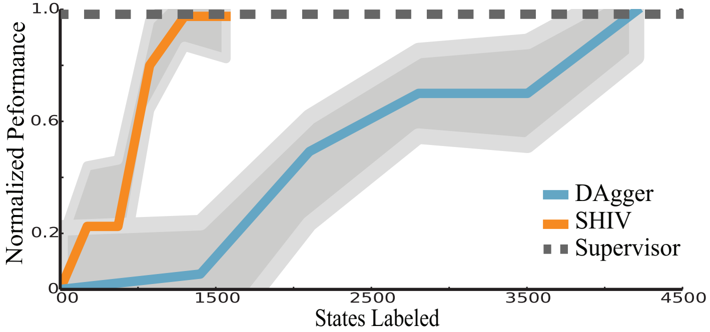




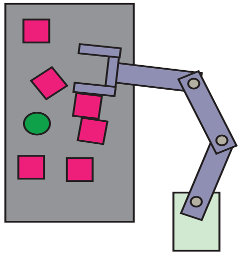
(c) Surgical Needle Insertion

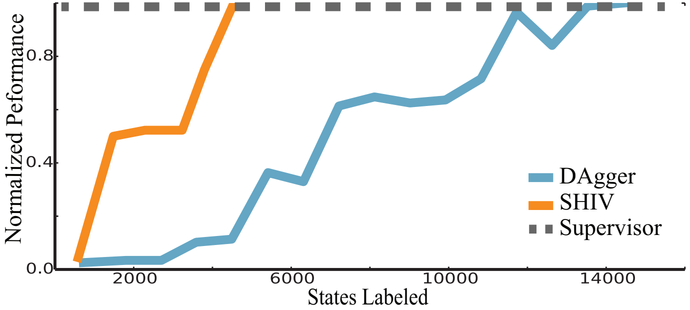
Slide from
Michael Laskey



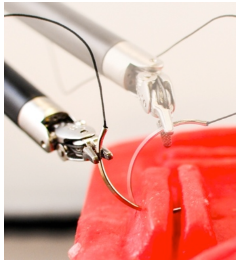


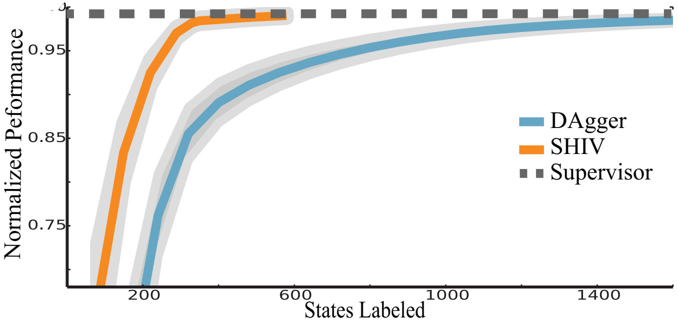
(a) Driving





(b) Grasping in Clutter in Box2D





(c) Surgical Needle Insertion

Slide from
Michael Laskey

Benefits and Limitations

- Substantially reduce human intervention
- Get safety of expert while still getting on-policy data
- Classifier may fail
  - Predicting accuracy is not easy
- Intermittent human intervention is not reliable

Safety without reliance on human to rescue the agent?

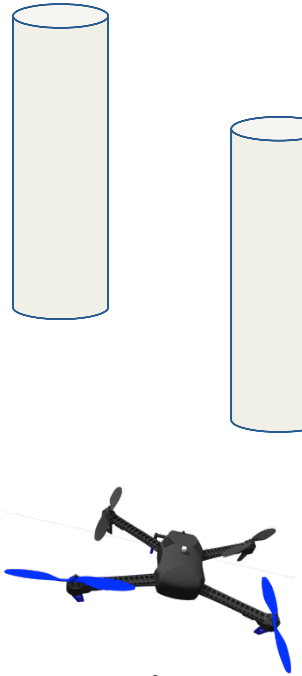
Case Study: Safety in RL

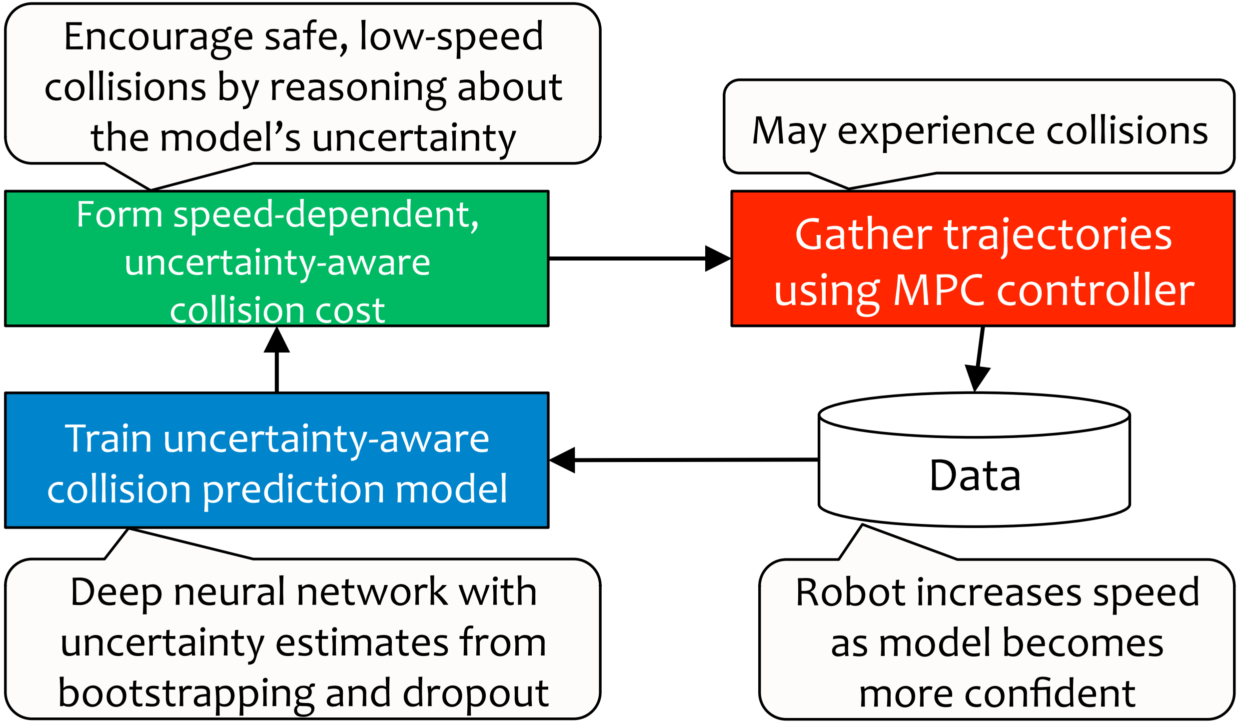
Uncertainty-Aware Reinforcement Learning for Collision Avoidance

Gregory Kahn*, Adam Villaflor*, Vitchyr Pong*, Pieter Abbeel*†, Sergey Levine*
*Berkeley AI Research (BAIR), University of California, Berkeley
†OpenAI

Approach

- Enable autonomous agents to safely act in complex, a priori unknown environments





Slide adapted from Greg Kahn

Collision prediction model

collision

command velocities

neural network $\boxed{p(\boxed{c_{t+H}} \mid \boxed{\mathbf{o}_t}, \boxed{\mathbf{u}_t, \ldots, \mathbf{u}_{t+H}})}$

raw image

Slide adapted from Greg Kahn

Uncertainty-aware collision cost

$$c_{\text{COLL}}(\tau) \propto \text{SPEED} \cdot \left(\mathrm{E}[p(c_{t+H}|\tau)] + \sqrt{\mathrm{Var}[p(c_{t+H}|\tau)]}\right)$$

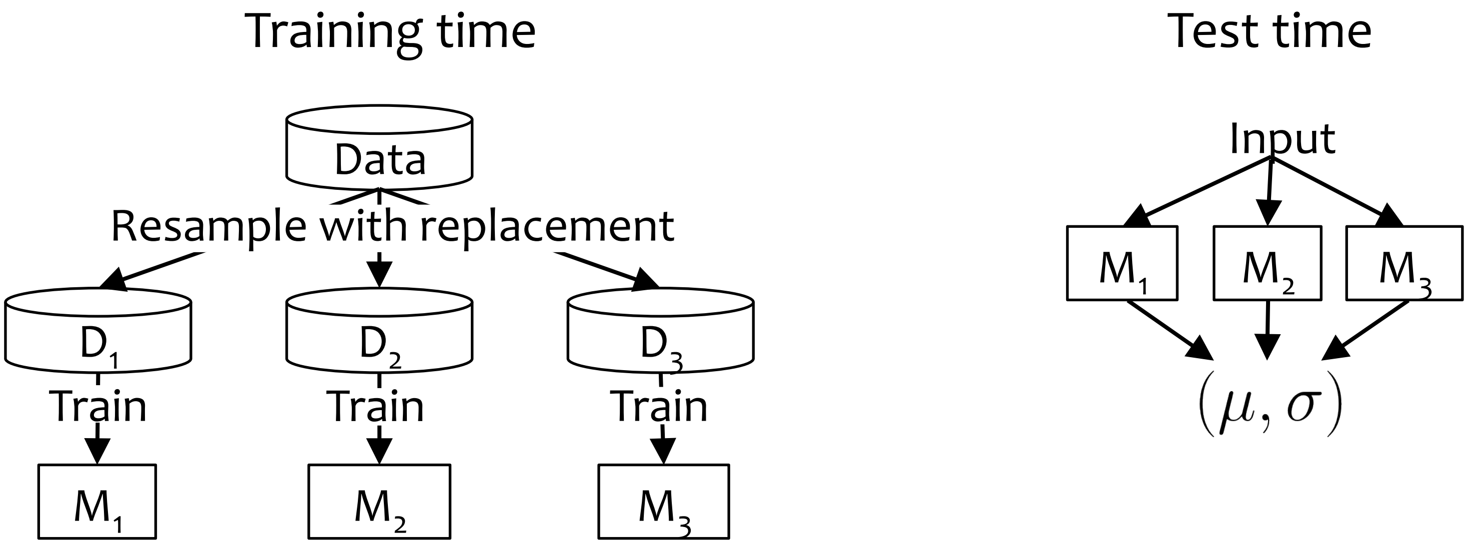
high speed   predict collision   large uncertainty

large cost

Slide adapted from Greg Kahn

Estimating neural network output uncertainty

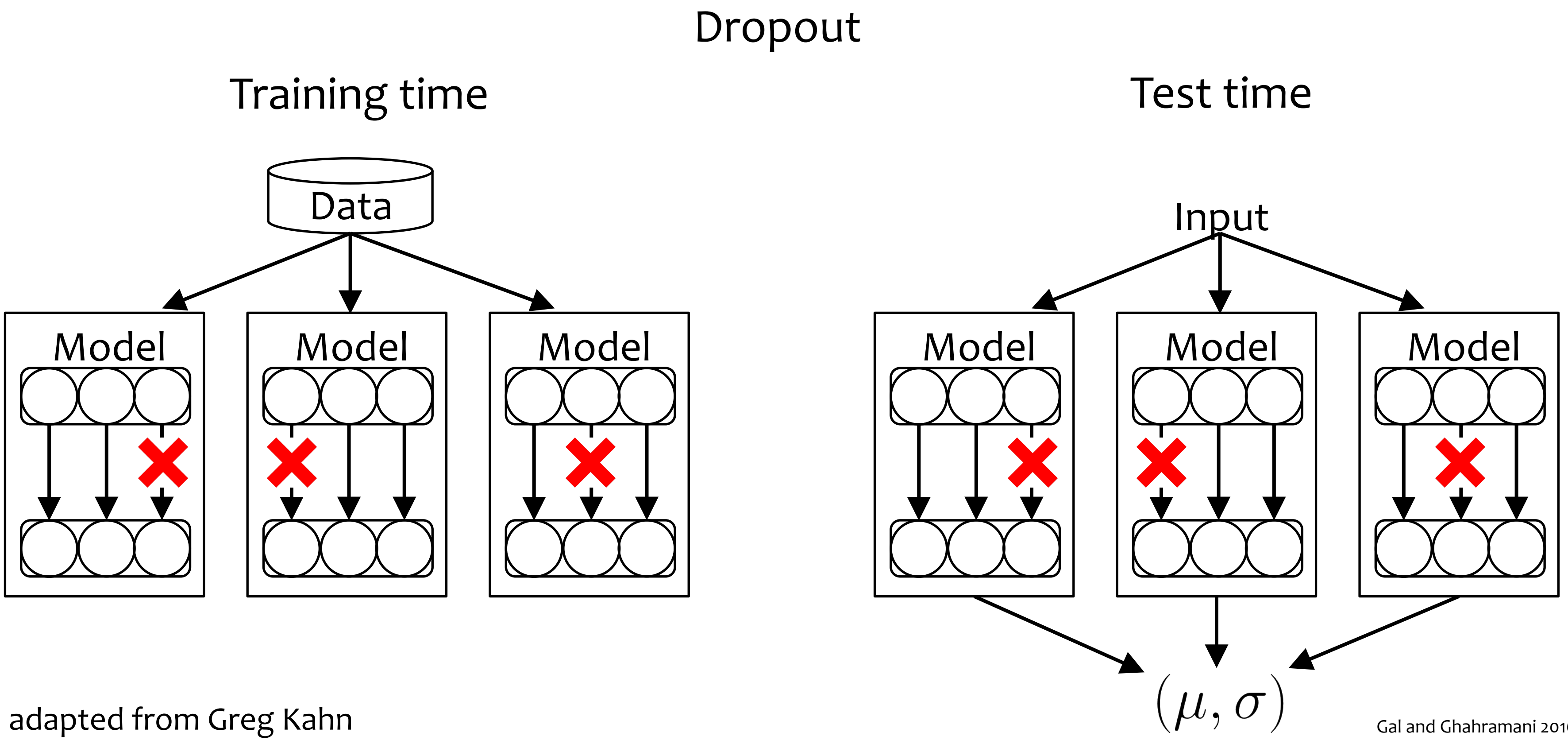
Bootstrapping



Slide adapted from Greg Kahn

Efron and Tibshirani 1982

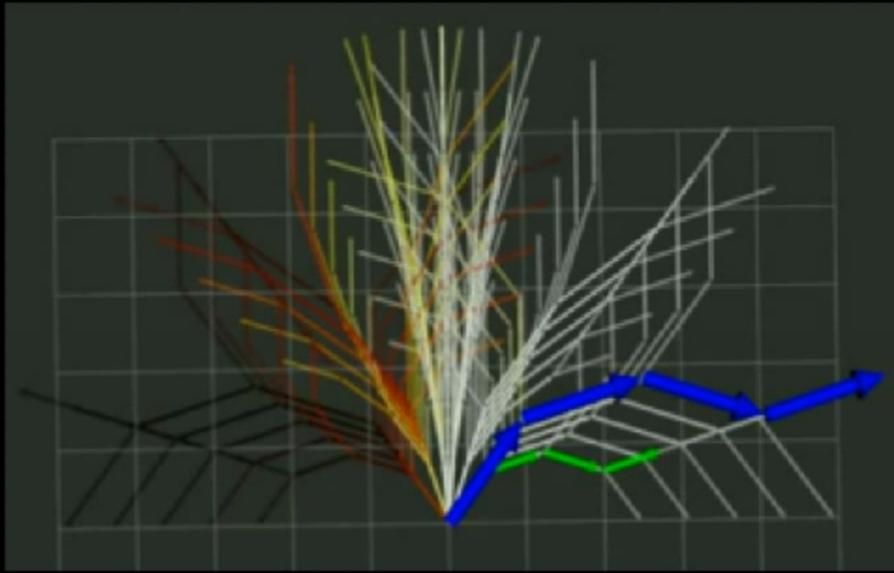
Estimating neural network output uncertainty



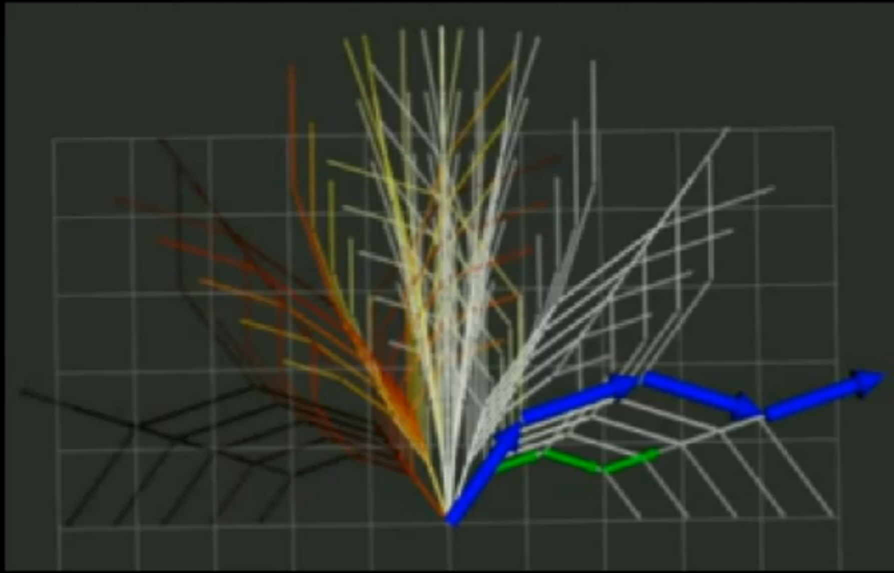
Slide adapted from Greg Kahn

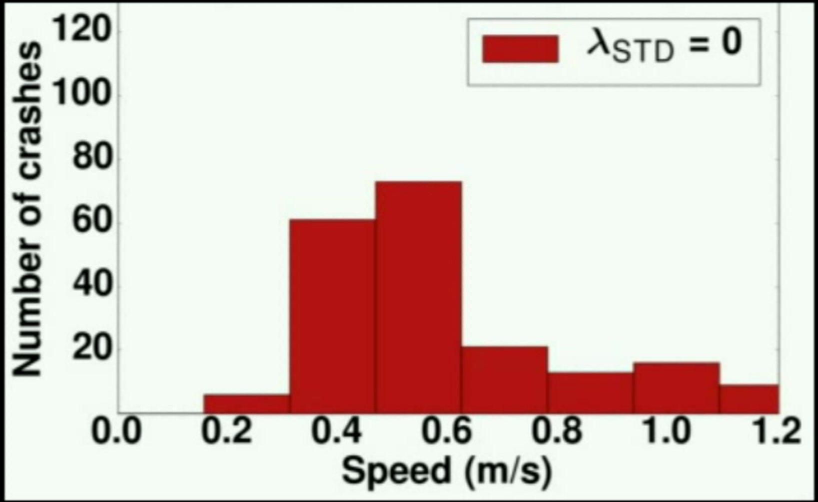
Gal and Ghahramani 2016


UNITED STATES
POSTAL SERVICE
PROF
LEVINE
450
SDH

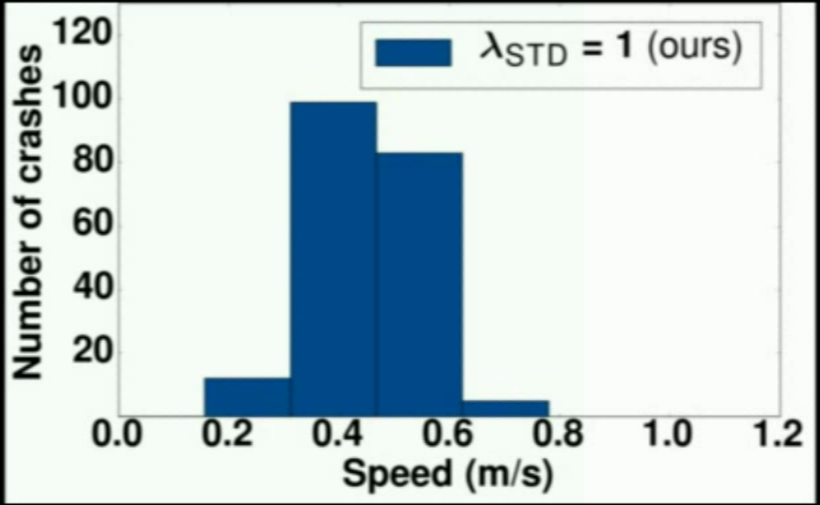



Initial random policy

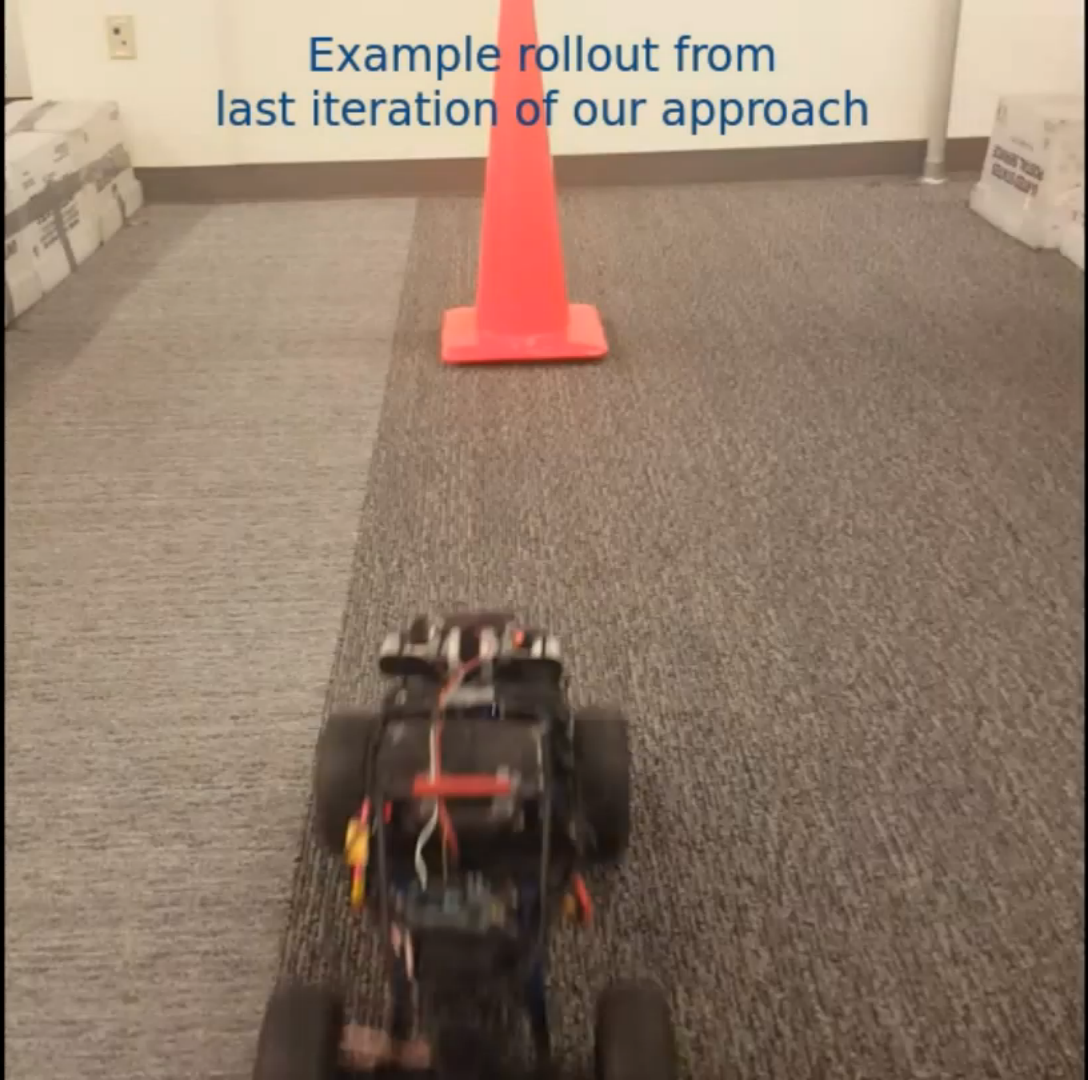


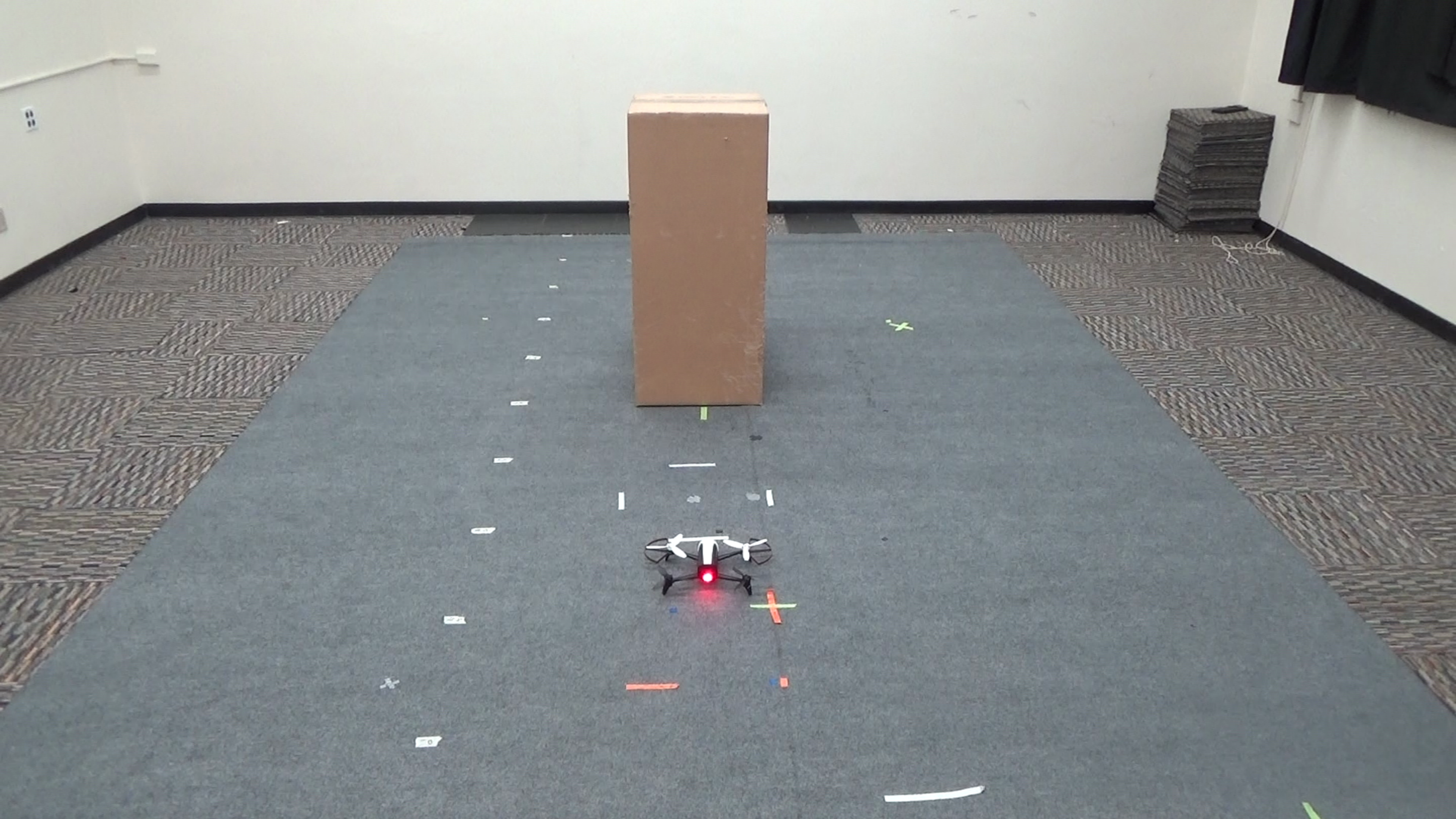


λ_{STD} = 0
Number of crashes
120
100
80
60
40
20
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Speed (m/s)

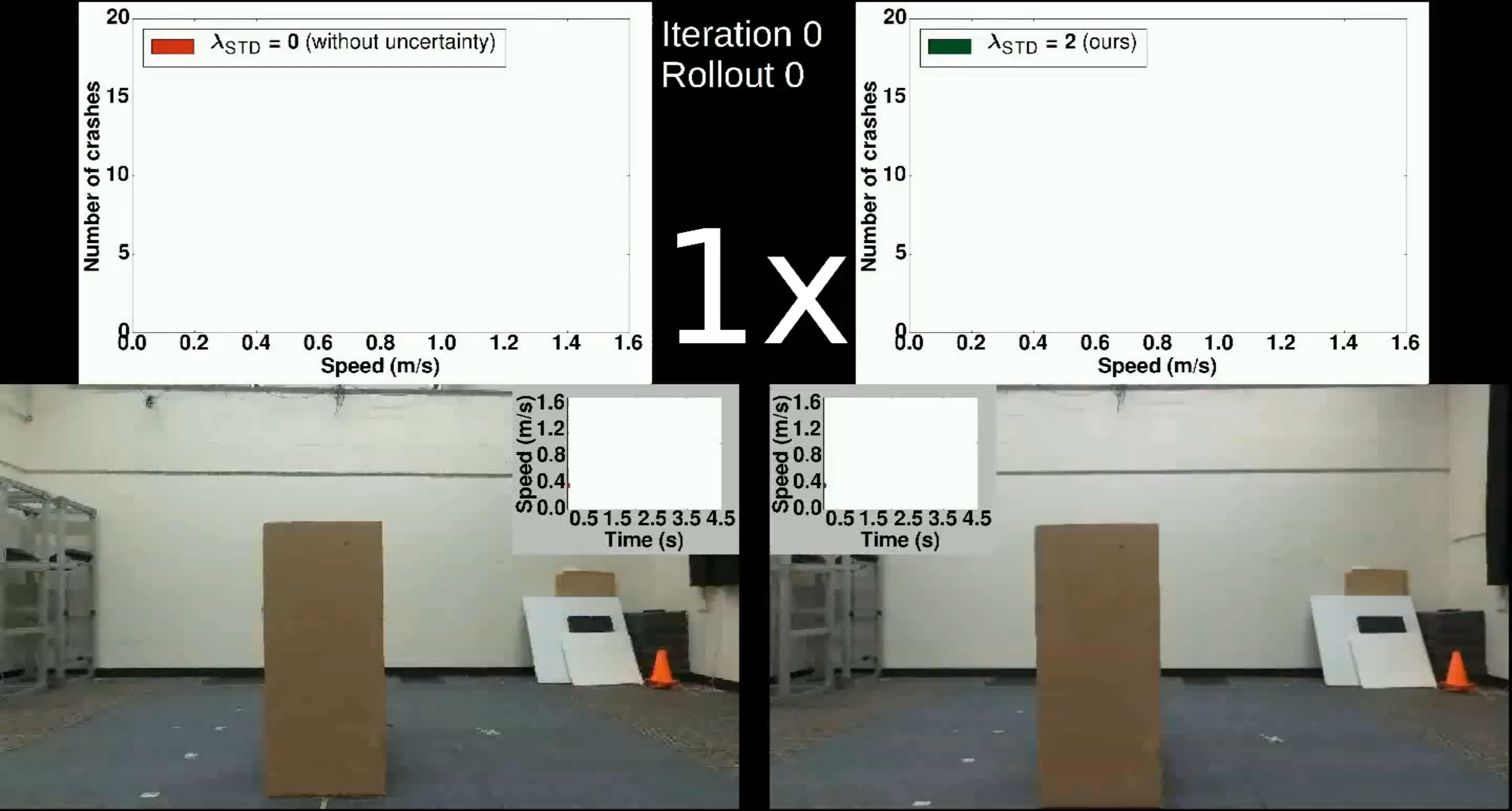


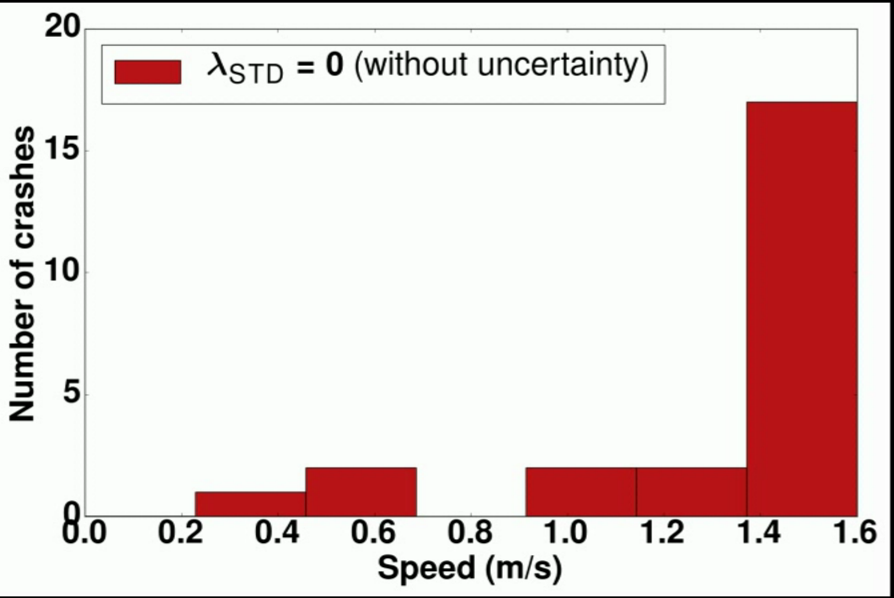
λ_{STD} = 1 (ours)
Number of crashes
120
100
80
60
40
20
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Speed (m/s)


Example rollout from
last iteration of our approach

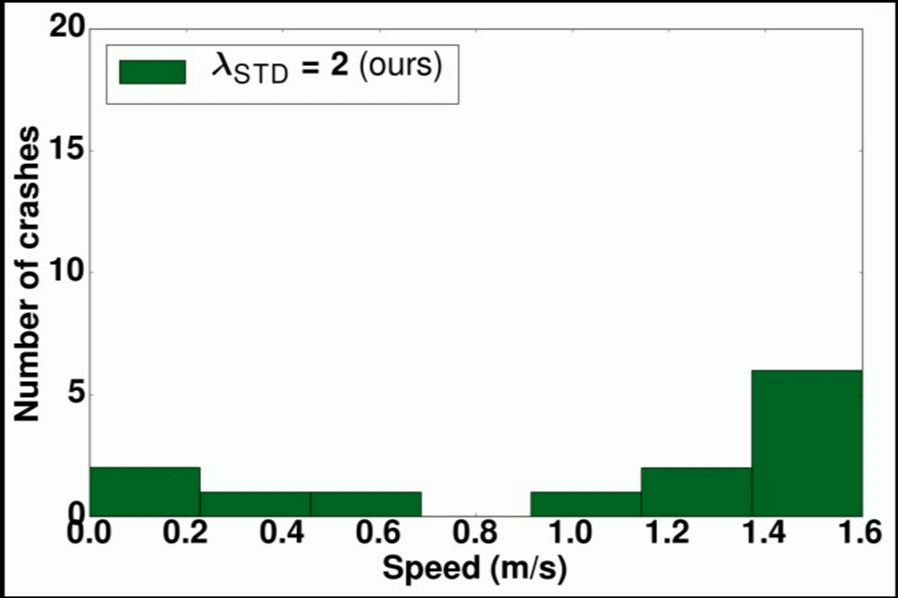


Safe versus unsafe collisions


λ_{STD} = 0 (without uncertainty)
Number of crashes
20
15
10
5
0
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6
Speed (m/s)
Iteration 0
Rollout 0
1x
λ_{STD} = 2 (ours)
Number of crashes
20
15
10
5
0
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6
Speed (m/s)
Speed (m/s)
1.6
1.2
0.8
0.4
0.0
0.5 1.5 2.5 3.5 4.5
Time (s)
Speed (m/s)
1.6
1.2
0.8
0.4
0.0
0.5 1.5 2.5 3.5 4.5
Time (s)



λ_{STD} = 0 (without uncertainty)
Number of crashes
Speed (m/s)



λ_{STD} = 2 (ours)
Number of crashes
Speed (m/s)

Example rollout from last iteration of our approach

Benefits and Limitations

- Take into account uncertainty of collision!
- No reliance on human to take control
- Slowing down might not always be a safe option
- Getting good uncertainty estimates is hard

Safety– Conclusions

To learn safely:

- Learn using data from a safe policy (off-policy)
- Account for uncertainty
- Be cautious (e.g. slow) in high-risk situations

Open Challenge:

- Predicting safety, and model's uncertainty

Advanced Topics in Imitation & Safety

1. Imitating humans: handling domain shift
2. Safety while learning
3. **Improving imitation learning from experts**

Recap: Guided Policy Search

$$\min_{\tau,\theta} c(\tau) \text{ s.t. } \mathbf{u}_t = \pi_\theta(\mathbf{x}_t)$$

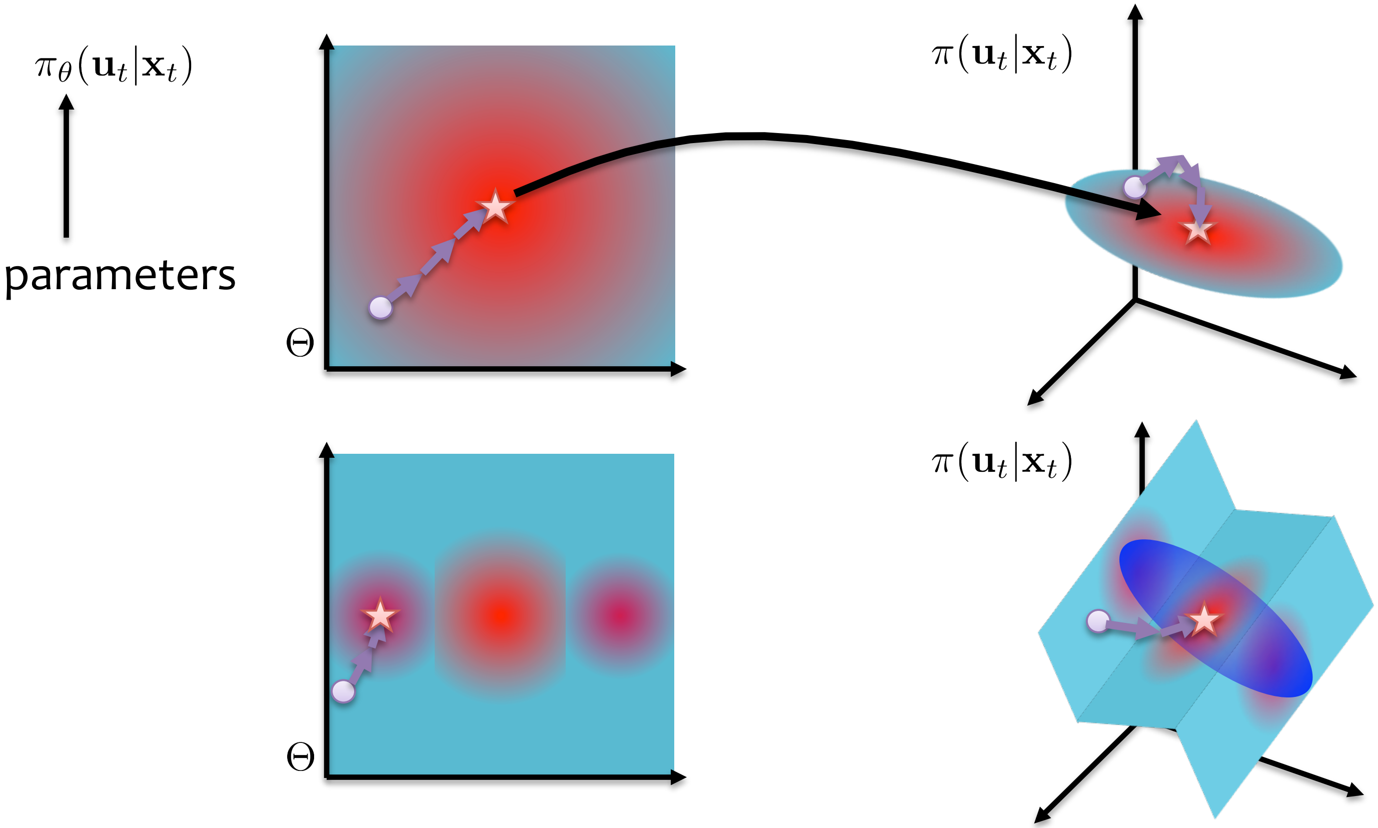
$$\bar{\mathcal{L}}(\tau, \theta, \lambda) = c(\tau) + \sum_{t=1}^{T} \lambda_t(\pi_\theta(\mathbf{x}_t) - \mathbf{u}_t) + \sum_{t=1}^{T} \rho_t(\pi_\theta(\mathbf{x}_t) - \mathbf{u}_t)^2$$

1. Find $\tau \leftarrow \arg\min_\tau \bar{\mathcal{L}}(\tau, \theta, \lambda)$ (e.g. via iLQR)
2. Find $\theta \leftarrow \arg\min_\theta \bar{\mathcal{L}}(\tau, \theta, \lambda)$ (e.g. via SGD)
3. $\lambda \leftarrow \lambda + \alpha \frac{dg}{d\lambda}$

Recap: Policy Gradient

$$\nabla_\theta \mathbb{E}_\tau [R] \approx \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} - b(s_t) \right) \right]$$

Parameter Space vs Policy Space



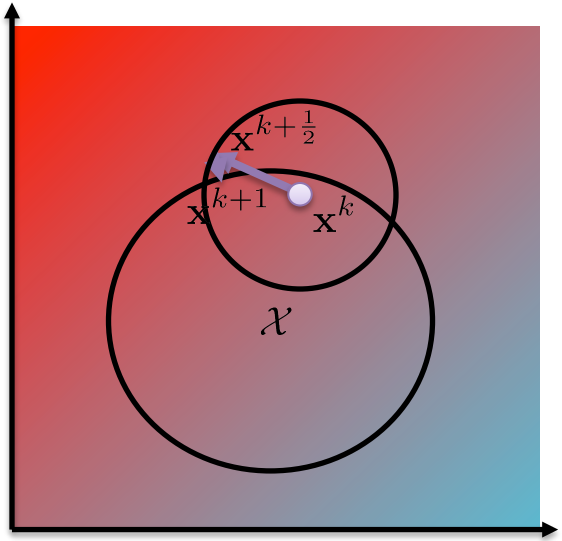
why policy space?

- local optima/ easier optimization landscapes
- can be **easier** to update in policy space vs parameter space

Slide from Sergey Levine

Montgomery & Levine '16

Mirror Descent Guided Policy Search (MDGPS)



$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{X}$$

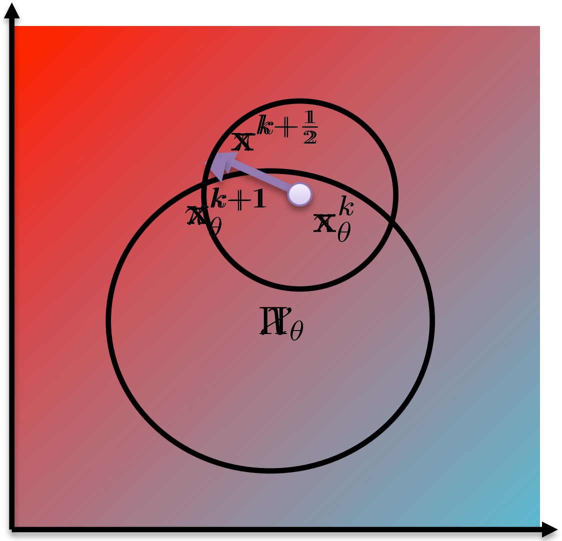
$$\mathbf{x}^{k+\frac{1}{2}} \leftarrow \min_{\mathbf{x}} \hat{f}(\mathbf{x}) \text{ s.t. } D(\mathbf{x}, \mathbf{x}^k) \leq \epsilon$$

$$\mathbf{x}^{k+1} \leftarrow \min_{\mathbf{x}} D(\mathbf{x}, \mathbf{x}^{k+\frac{1}{2}}) \text{ s.t. } \mathbf{x} \in \mathcal{X}$$

Slide from Sergey Levine

Montgomery & Levine '16

Mirror Descent Guided Policy Search (MDGPS)



$$\min_{\mathbf{x}} J(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \Pi_\theta$$

$$\mathbf{x}^{k+\frac{1}{2}} \leftarrow \min_{\mathbf{x}} \hat{J}(\mathbf{x}) \text{ s.t. } D(\mathbf{x}, \mathbf{x}_\theta^k) \leq \epsilon$$

$$\mathbf{x}_\theta^{k+1} \leftarrow \min_{\theta} D(\mathbf{x}_\theta, \mathbf{x}^{k+\frac{1}{2}}) \text{ s.t. } \mathbf{x} \in \mathcal{X}$$

“projection”: supervised learning

local policy optimization:

- trajectory-centric model-based RL [Montgomery ‘16]
- path integral policy iteration [Chebotar ‘16]

Slide from Sergey Levine

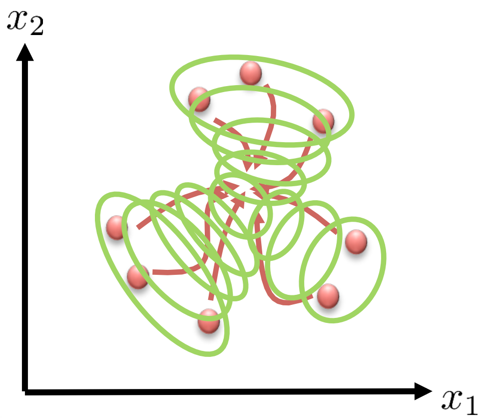
Montgomery & Levine ‘16

MDGPS with Random Initial States and Local Models

$$\min_{\pi} J(\pi) \text{ s.t. } \pi \in \Pi_\theta$$

$$\pi^{k+\frac{1}{2}} \leftarrow \min_{\pi} \hat{J}(\pi) \text{ s.t. } D_{\text{KL}}(\pi \| \pi_\theta^k) \leq \epsilon$$

$$\pi_\theta^{k+1} \leftarrow \min_{\theta} D_{\text{KL}}(\pi_\theta \| \pi^{k+\frac{1}{2}})$$



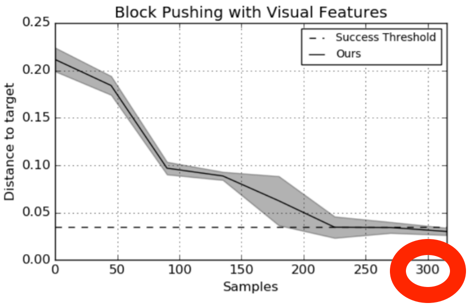
1. Fit N Gaussian trajectory distributions $p_i(\tau)$
2. For each distribution fit $p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ as time-varying linear-Gaussian.
3. Update time-varying linear-Gaussian $\pi_i(\mathbf{u}_t|\mathbf{x}_t)$ using LQR with KL constraint.

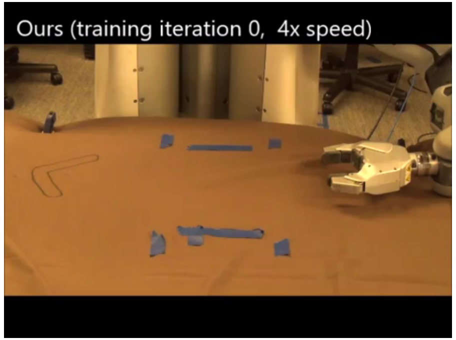
Use supervised learning to train neural net $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$ to mimic all N "local policies" $\pi_i(\mathbf{u}_t|\mathbf{x}_t)$

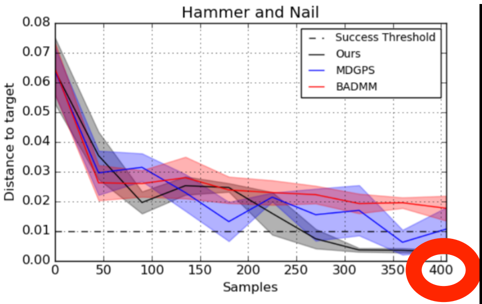
Montgomery*, Ajay*, et al. '17

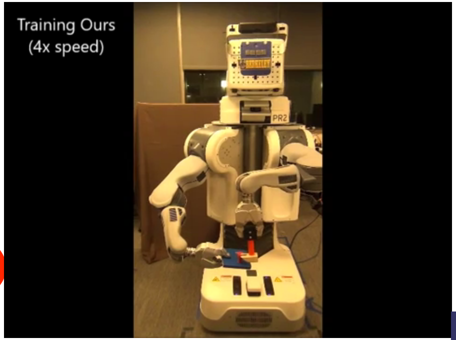
Slide from Sergey Levine

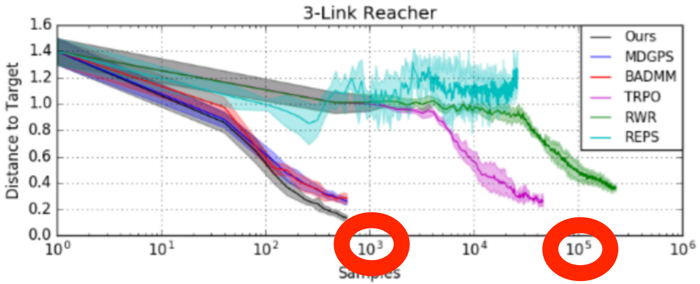
Efficiency & Real-World Evaluation

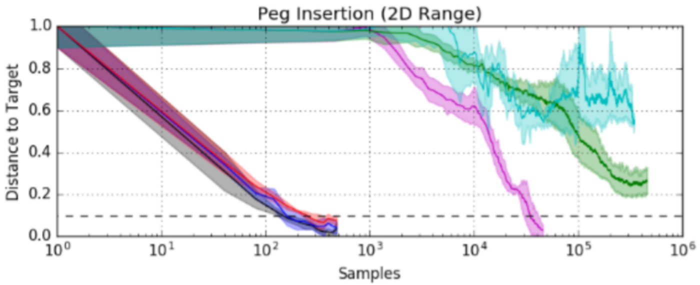


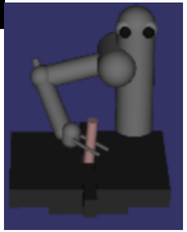


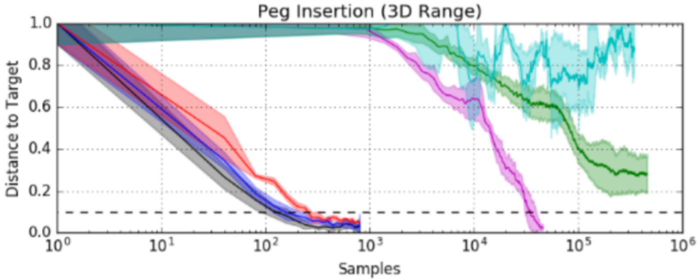












Learning 2D reaching
(simple benchmark task):

- TRPO (best known value): 3000 trials
- DDPG, NAF (best known value): 2000 trials
- Q-Prop: 2000 trials
- MDGPS: **500** trials

Slide from Sergey Levine

Imitation from Experts – Conclusions

- Optimization in policy space can be easier than in parameter space
- Use clustering for learning local-linear models with random initial states

Next time: Inverse Reinforcement Learning