Class notes

1. Homework 5 due Tuesday, November 13th 11:59pm

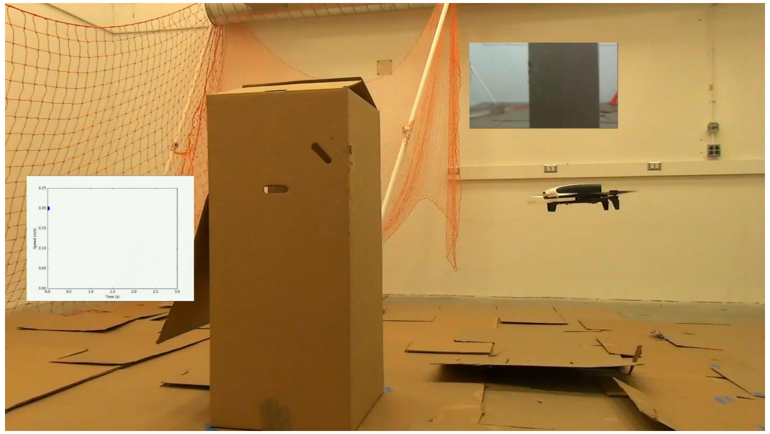
Real-World Robot Learning: Safety and Flexibility

CS294-112: Deep Reinforcement Learning

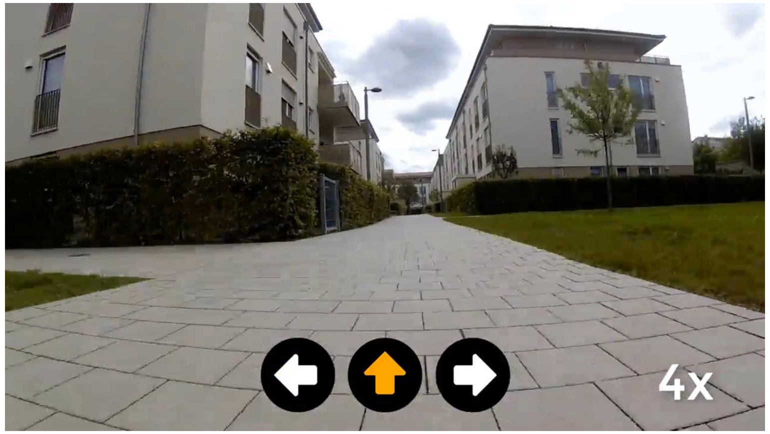
Gregory Kahn

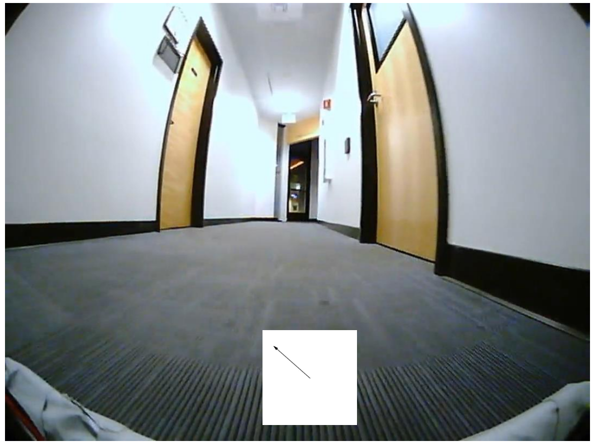
Why should you care?

Safety



Flexibility





Outline

Topics

- Safety
- Flexibility

Algorithms

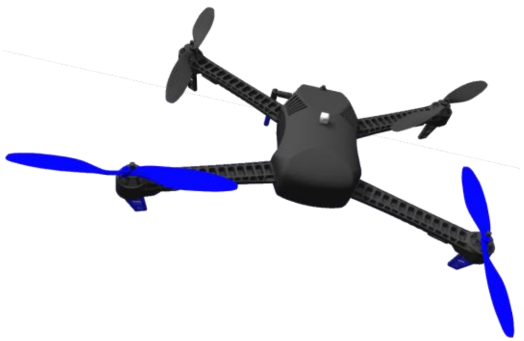
- Imitation learning
- Model-free
- Model-based

2 * 3 = 6 papers we'll cover

By no means the best / only papers on these topics

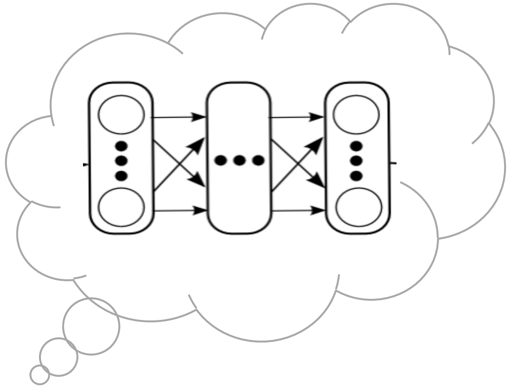
PLATO: Policy Learning using Adaptive Trajectory Optimization

Gregory Kahn[1], Tianhao Zhang[1], Sergey Levine[1], Pieter Abbeel[1,2,3]



Goal

Learn control policy that **maps observations to controls**

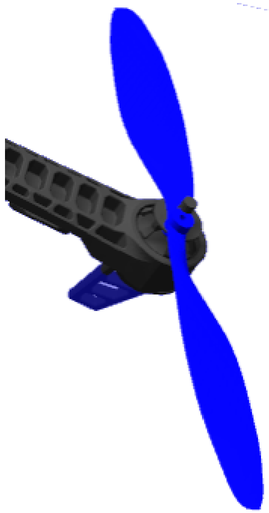


Control $\mathbf{u}_t$

Observation $\mathbf{o}(\mathbf{x}_t)$

Policy

$\pi_\theta(\mathbf{o}(\mathbf{x}_t))$



Assumption

- Able to generate good trajectories using an expert policy π^*

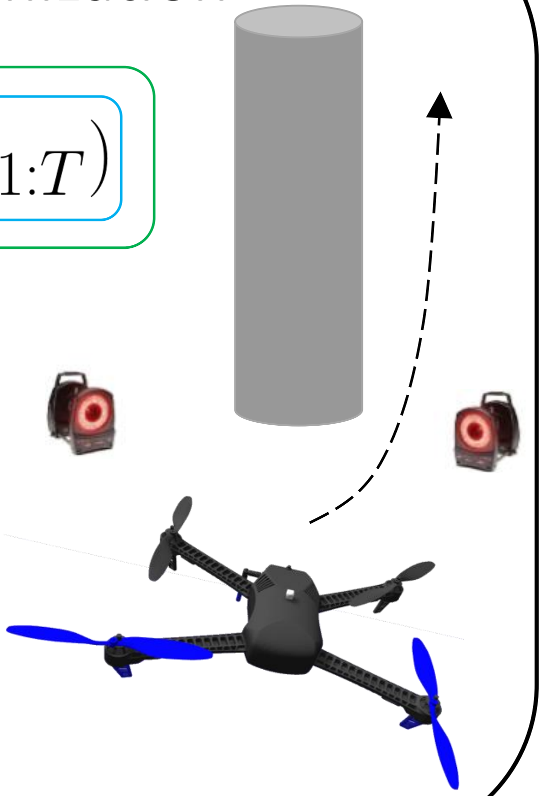
Human expert

Trajectory optimization

$$\boxed{c(\boxed{\mathbf{x}_0}, \mathbf{u}_{1:T})}$$

- cost function
- optimization
- full state information

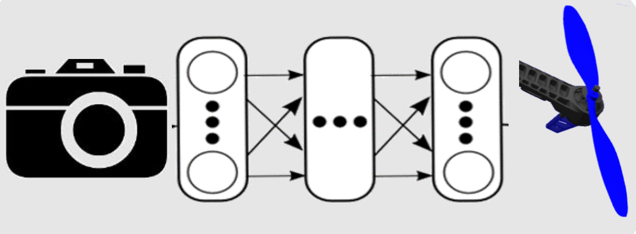
only during training



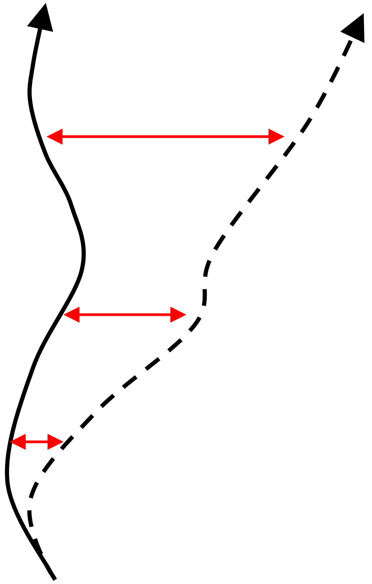
Supervised Learning

Trajectory optimization

Gather expert trajectories → Supervised learning



Training trajectory



Learned policy trajectory

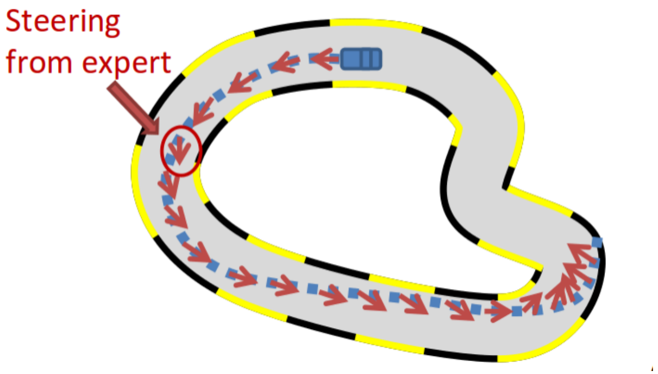
Policy reaches states not in training set! [Ross et al 2010]

- Problem: training and test distributions differ

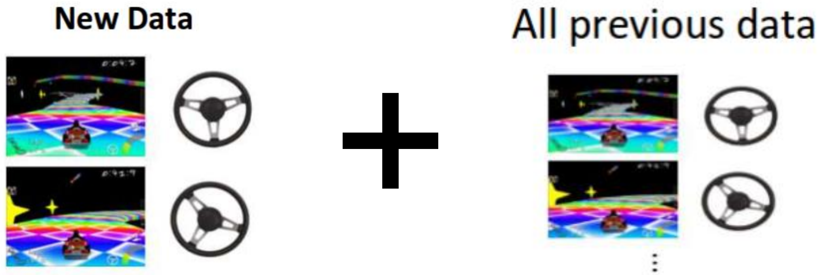
Dataset Aggregation (DAgger)

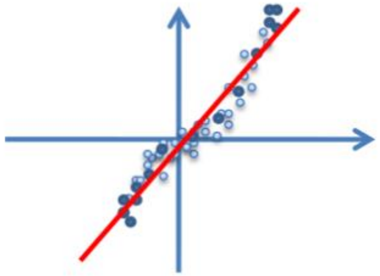
- Problem: training and test distributions differ
- Solution: execute policy during training

$$\pi_{\text{mix}} \leftarrow \beta\pi^* + (1-\beta)\pi_\theta$$



Gather expert trajectories → Supervised learning

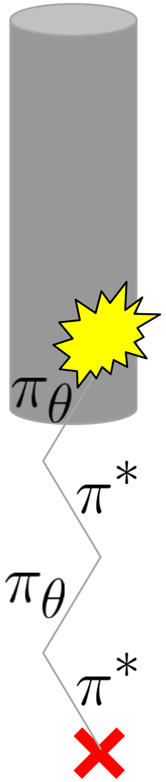




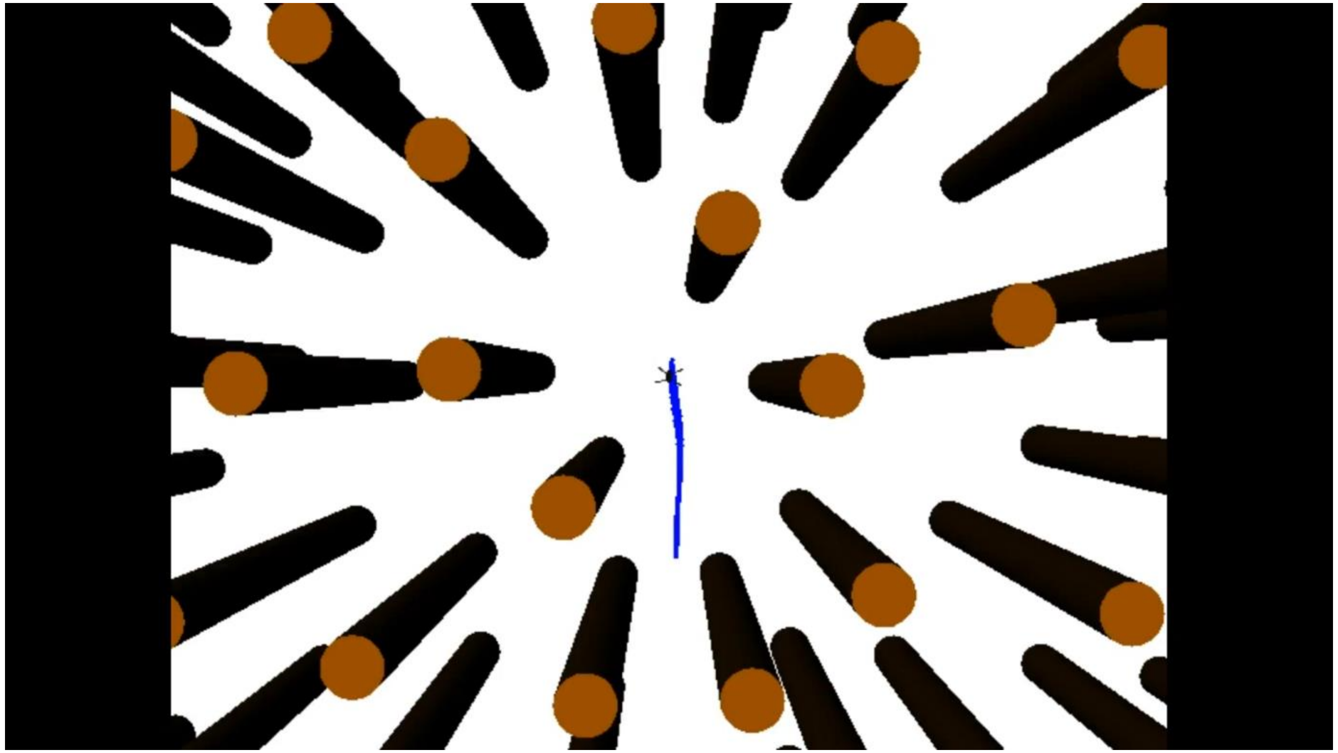
Safety during training

- DAgger mixes the actions

$$\mathbf{u}_{\text{mix}} \sim \begin{cases} \pi^*(\mathbf{u}|\mathbf{x}_t) & \text{prob. } \beta \\ \pi_\theta(\mathbf{u}|\mathbf{x}_t) & \text{prob. } (1-\beta) \end{cases}$$



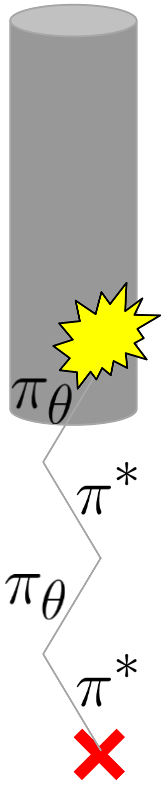
$$\beta = \frac{1}{2}$$



Policy Learning using Adaptive Trajectory Optimization (PLATO)

- DAgger mixes the actions

$$\mathbf{u}_{\text{mix}} \sim \begin{cases} \pi^*(\mathbf{u}|\mathbf{x}_t) & \text{prob. } \beta \\ \pi_\theta(\mathbf{u}|\mathbf{x}_t) & \text{prob. } (1-\beta) \end{cases}$$

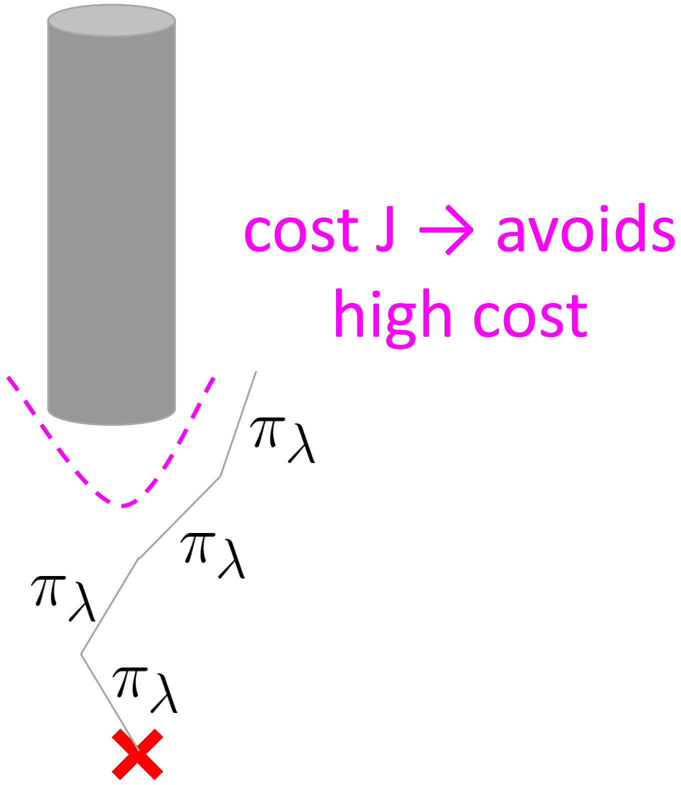


$$\beta = \frac{1}{2}$$

- PLATO mixes the objectives

$$\pi_\lambda \leftarrow \arg\min_\pi J(\pi) + \lambda D_{\text{KL}}(\pi || \pi_\theta)$$

$$\mathbf{u}_\lambda \sim \pi_\lambda(\mathbf{u}|\mathbf{x}_t)$$



Algorithm comparisons

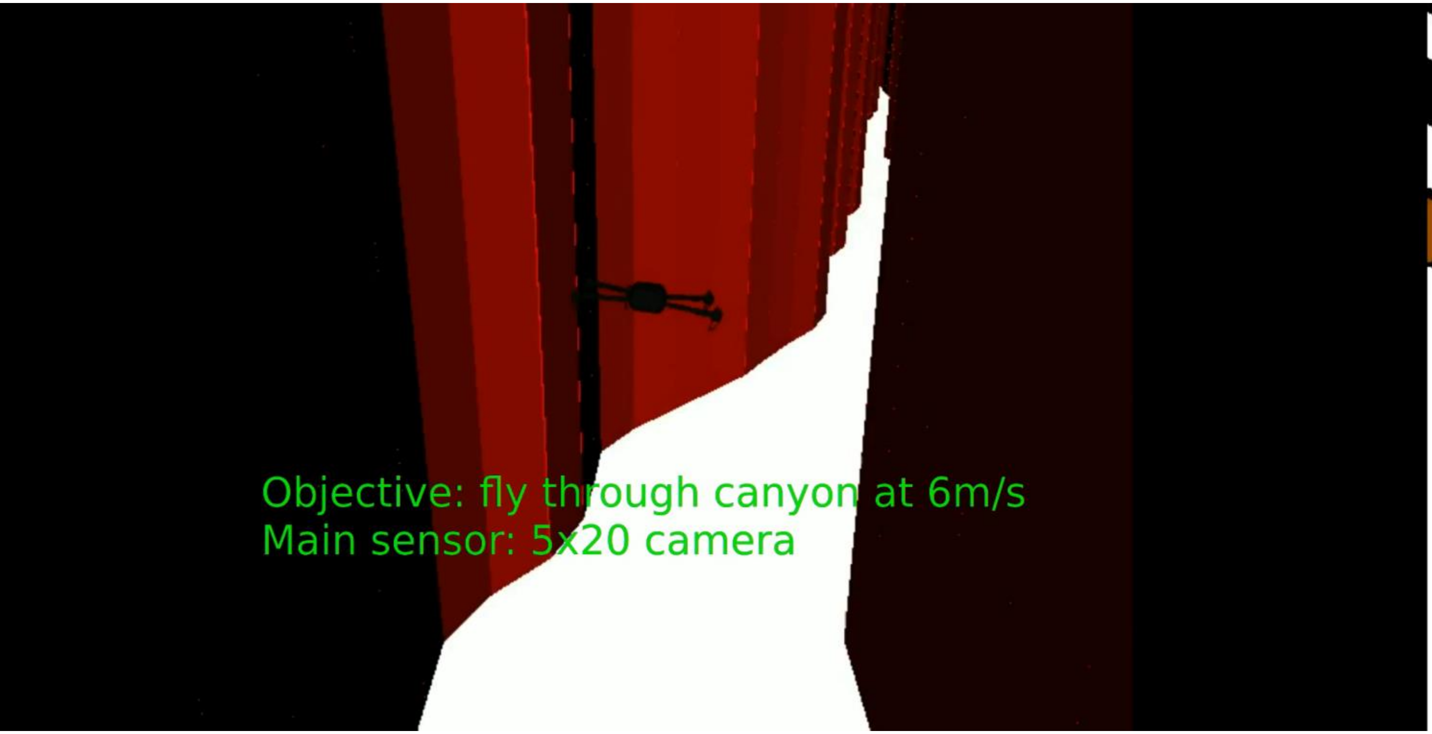
approach	sampling policy	safe	similar training and test distributions
supervised learning	π^*	✔	✘
DAgger	π_{mix}	✘	✔
PLATO	π_λ	✔	✔

$$\pi_{\text{mix}} \leftarrow \beta\pi^* + (1-\beta)\pi_\theta$$

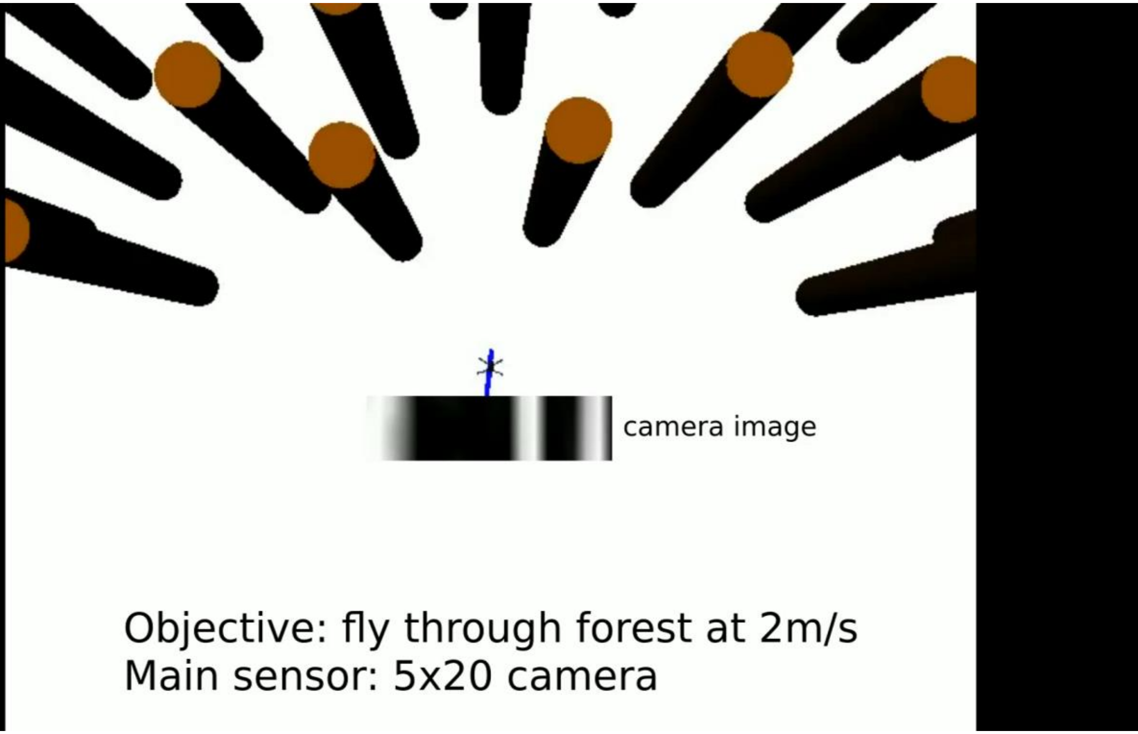
$$\pi_\lambda \leftarrow \arg\min_\pi J(\pi) + \lambda D_{\text{KL}}(\pi||\pi_\theta)$$

Experiments: final neural network policies

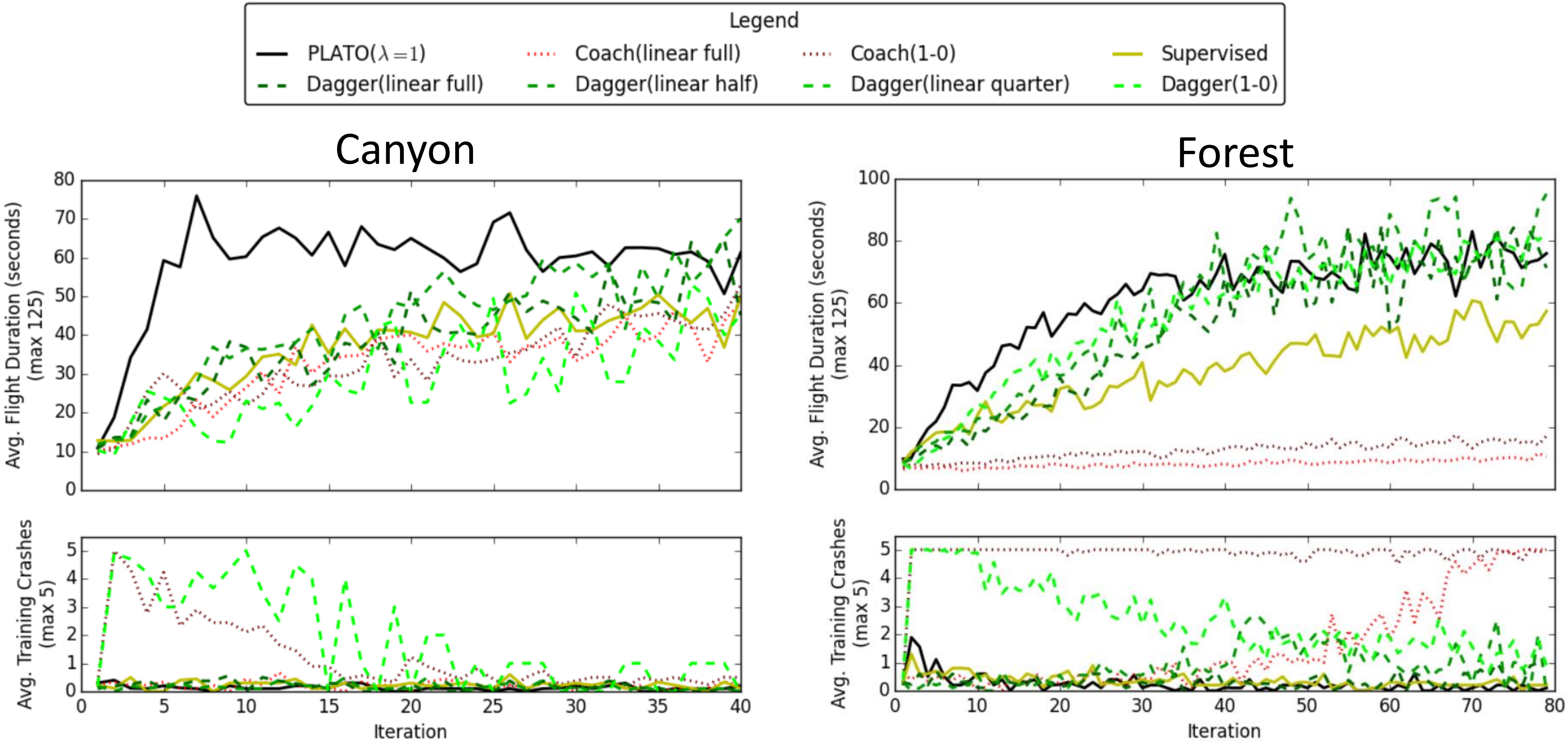
Canyon



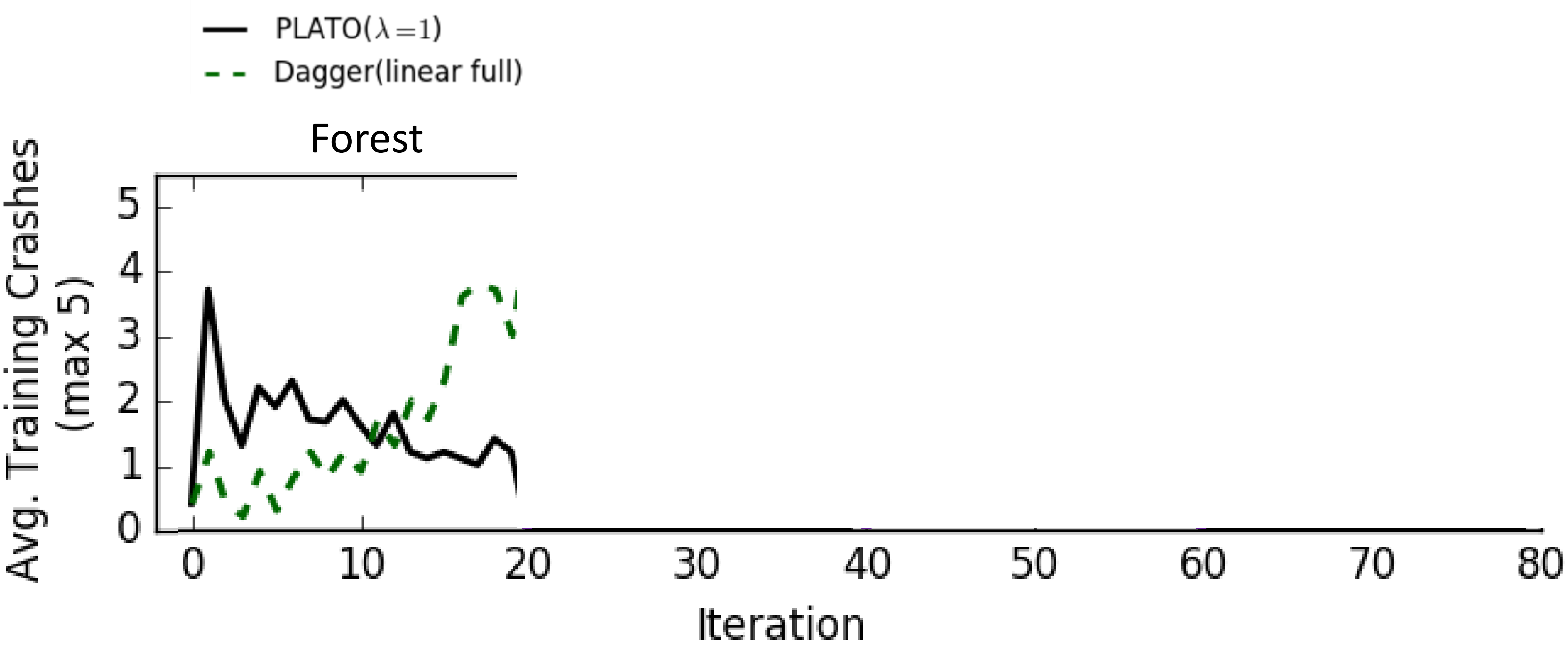
Forest



Experiments: metrics



Experiments: metrics

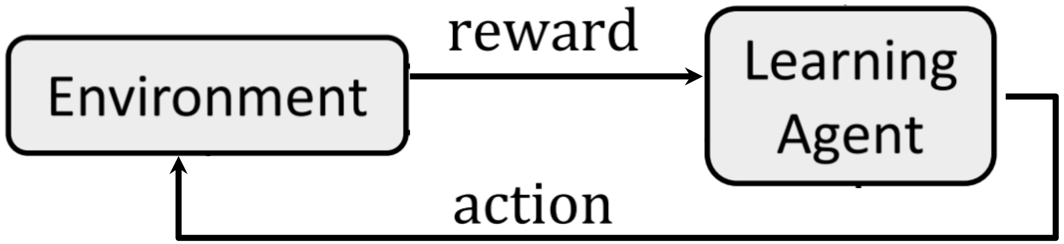


Safe Reinforcement Learning via Shielding

Mohammed Alshiekh[1], Roderick Bloem[2], Rüdiger Ehlers[3], Bettina Könighofer[2], Scott Niekum[1], Ufuk Topcu[1]

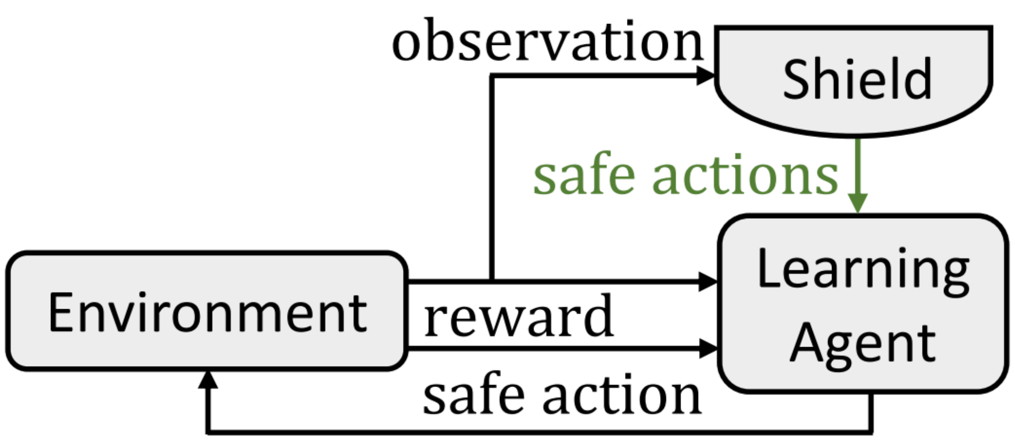
Goal

NOT SAFE



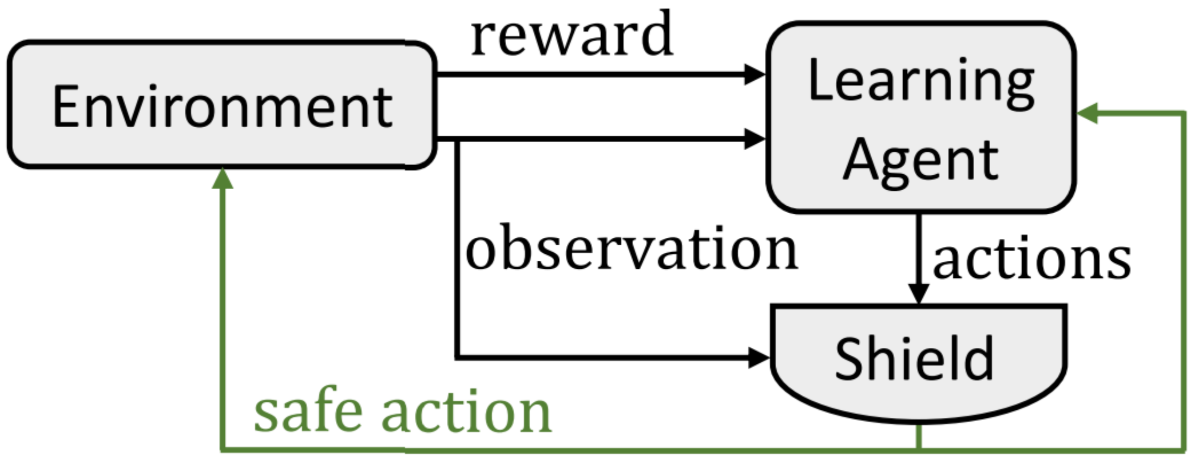
Shielding

Pre-emptive shielding



Like learning in a transformed MDP

Post-posed shielding

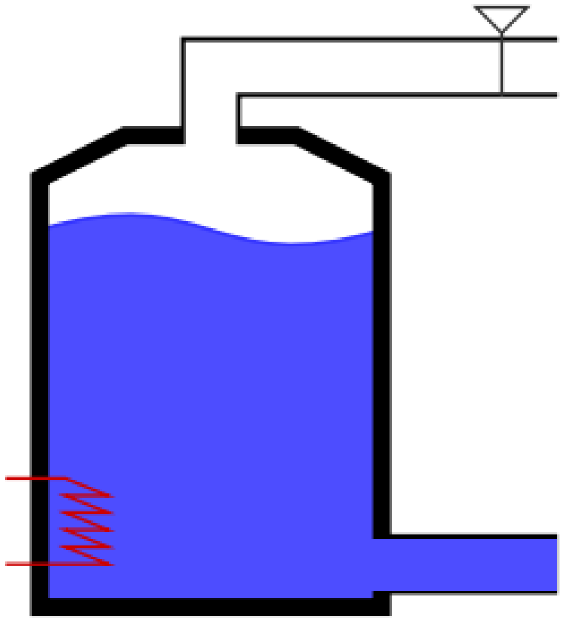


Shield can be used at test time

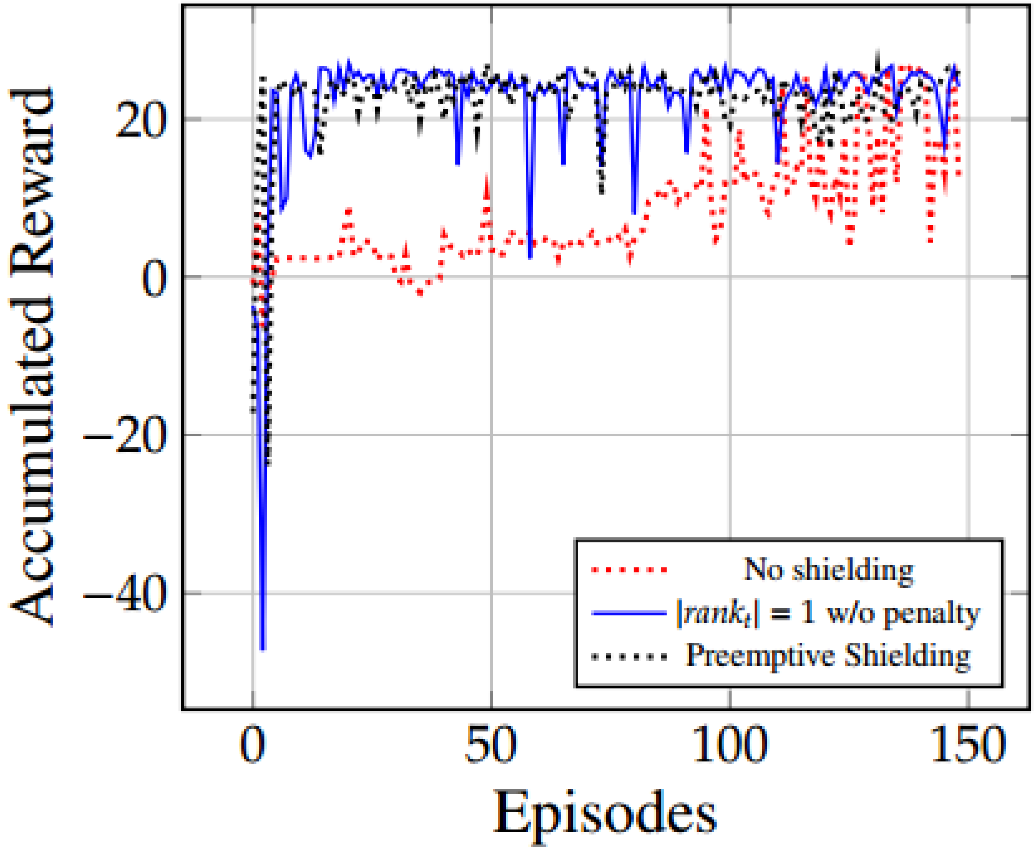
How to shield: linear temporal logic

- Encode safety with temporal logic
- Assumption: Known approximate/conservative transition dynamics

$$
\begin{aligned}
&\mathsf{G}(\mathit{level} > 0) \\
\wedge\ &\mathsf{G}(\mathit{level} < 100) \\
\wedge\ &\mathsf{G}((\mathit{open} \wedge \mathsf{X}\mathit{close}) \rightarrow \mathsf{XX}\mathit{close} \wedge \mathsf{XXX}\mathit{close}) \\
\wedge\ &\mathsf{G}((\mathit{close} \wedge \mathsf{X}\mathit{open}) \rightarrow \mathsf{XX}\mathit{open} \wedge \mathsf{XXX}\mathit{open})
\end{aligned}
$$

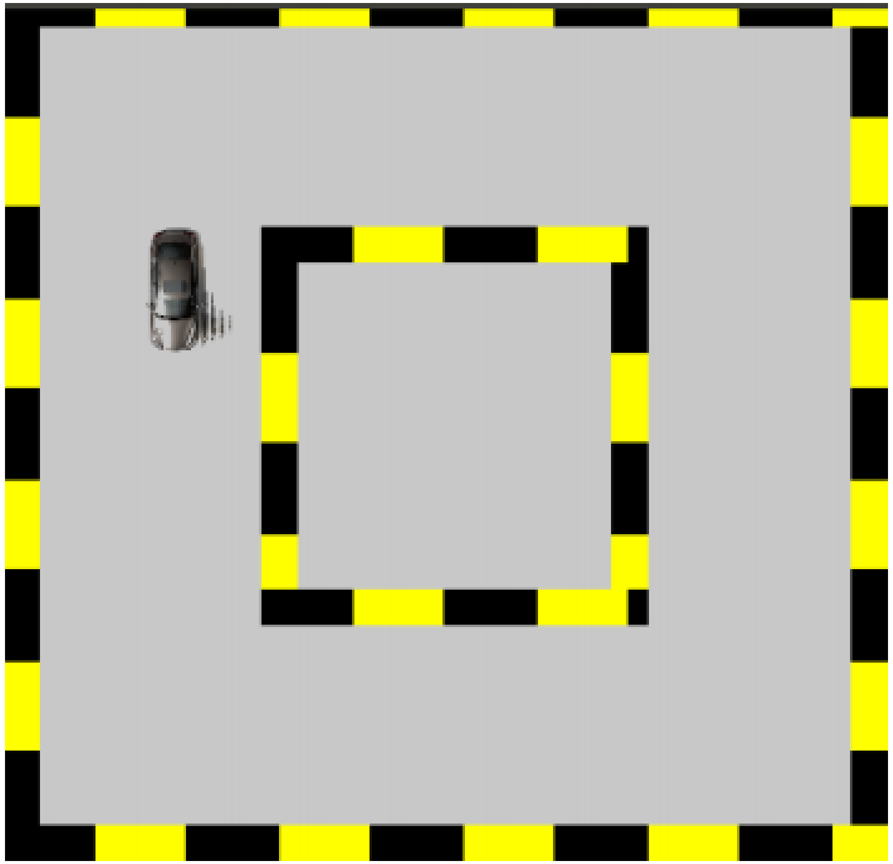


Experiments



Safety criteria

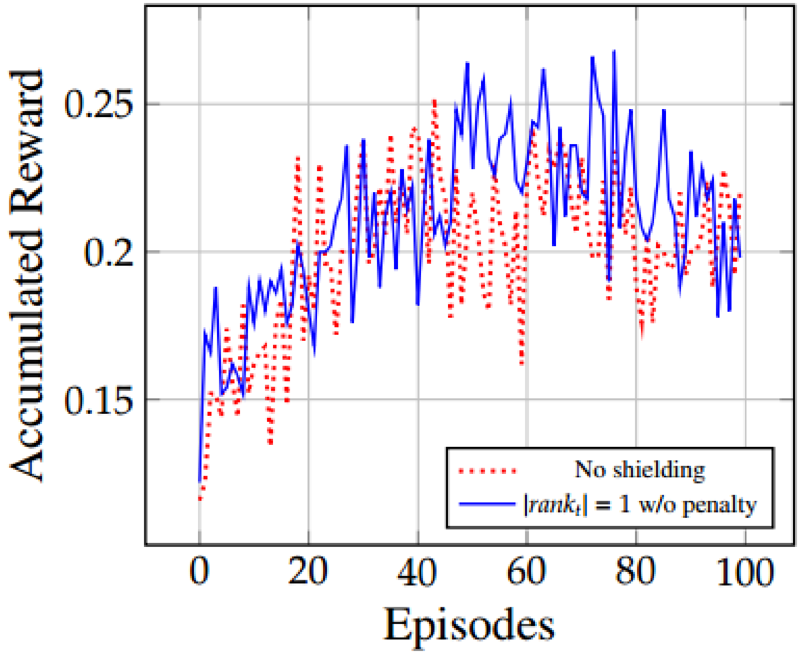
- Don’t crash



Experiments

Safety criteria

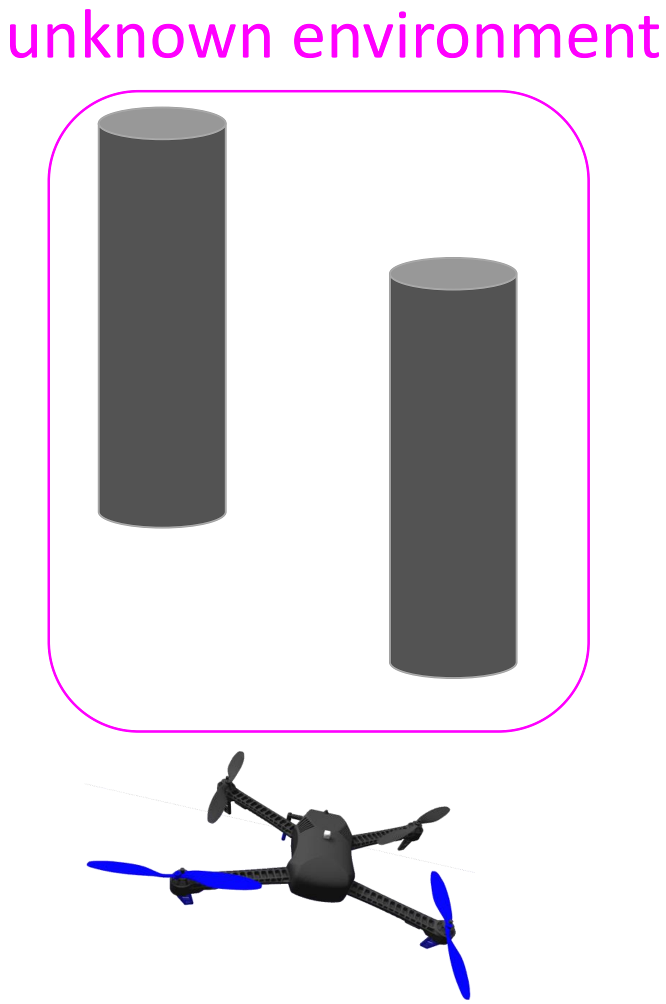
- Don't run out of oxygen
- If enough oxygen,
  don't surface w/o divers



Uncertainty-Aware Reinforcement Learning for Collision Avoidance

Gregory Kahn*, Adam Villaflor*, Vitchyr Pong*, Pieter Abbeel*†, Sergey Levine*

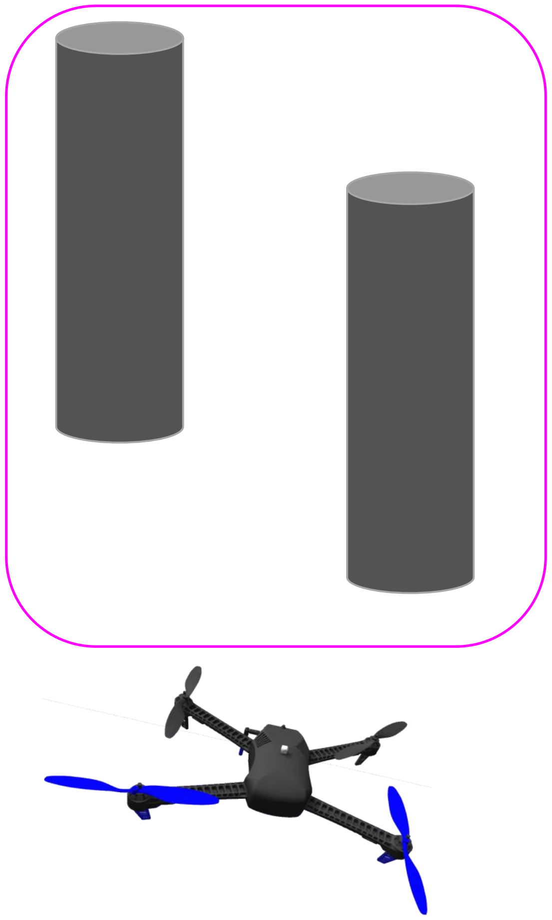
Goal



How to do reinforcement learning
without destroying the robot during training
using only onboard images

Approach

unknown environment



$$c(\tau) = c_{\text{TASK}}(\tau) + \boxed{c_{\text{COLL}}(\tau)}$$

learn a collision prediction model

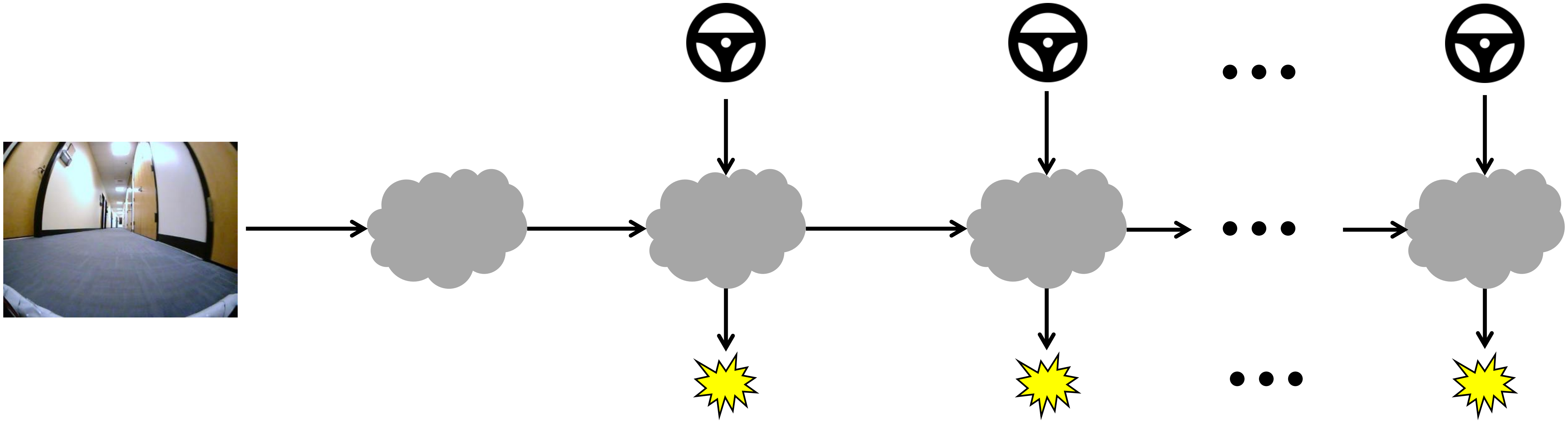
$$\boxed{p(c_{t+H} \mid \boxed{\mathbf{o}_t}, \boxed{\mathbf{u}_t, \ldots, \mathbf{u}_{t+H}})}$$

raw image

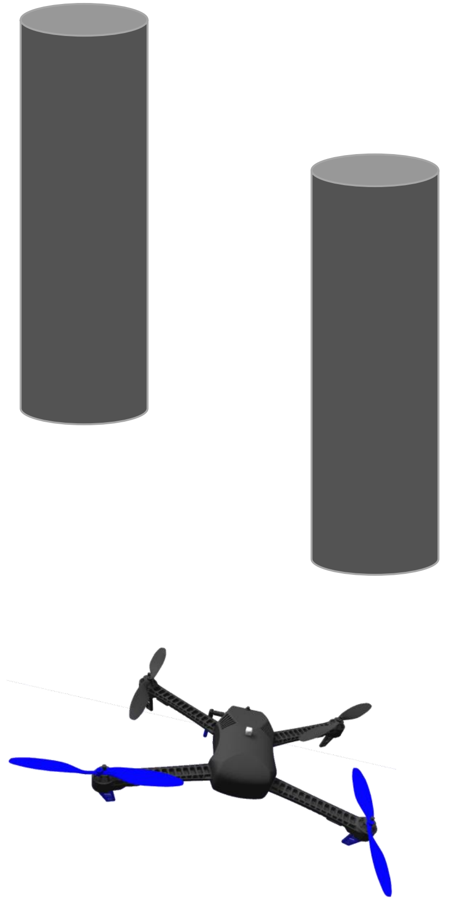
command velocities

neural network

Collision prediction model



Model-based RL using collision prediction model



Encourage safe, low-speed collisions by reasoning about the model's uncertainty

Form speed-dependent, uncertainty-aware collision cost $c_{\text{COLL}}(\tau)$

May experience collisions

Gather trajectories using MPC controller

Data

Robot increases speed as model becomes more confident

Train uncertainty-aware collision prediction model

Deep neural network with uncertainty estimates from bootstrapping and dropout

Collision cost

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

high speed

predict collision

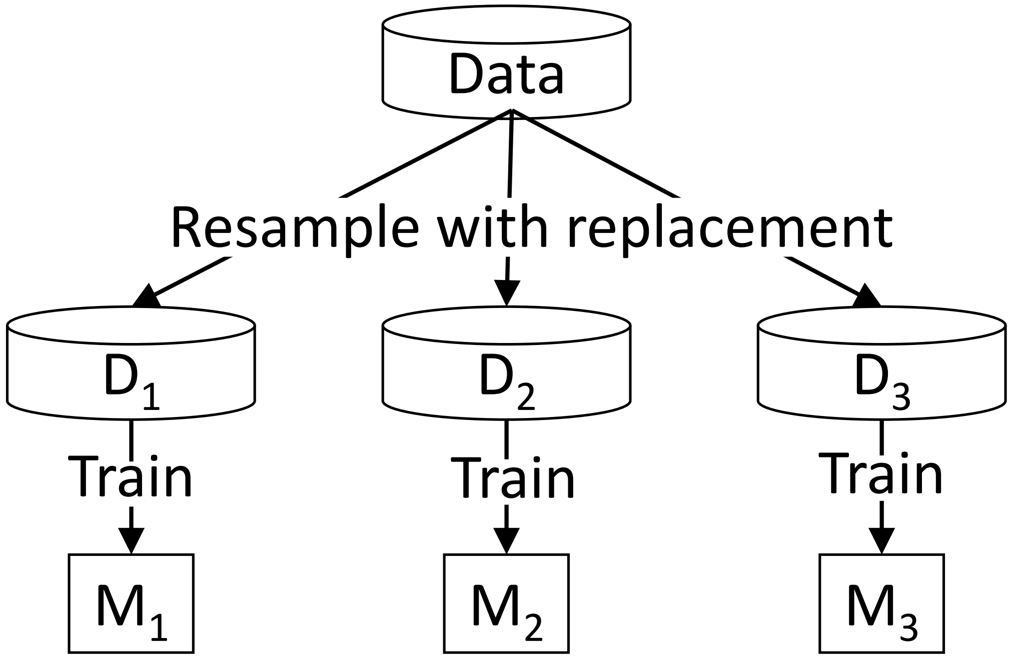
large uncertainty

large cost

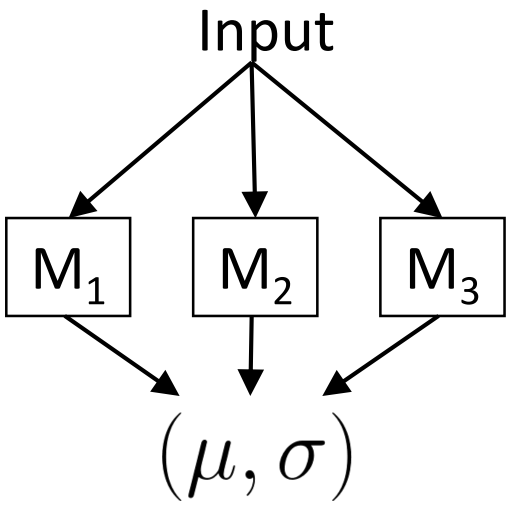
Estimating neural network output uncertainty

Bootstrapping

Training time

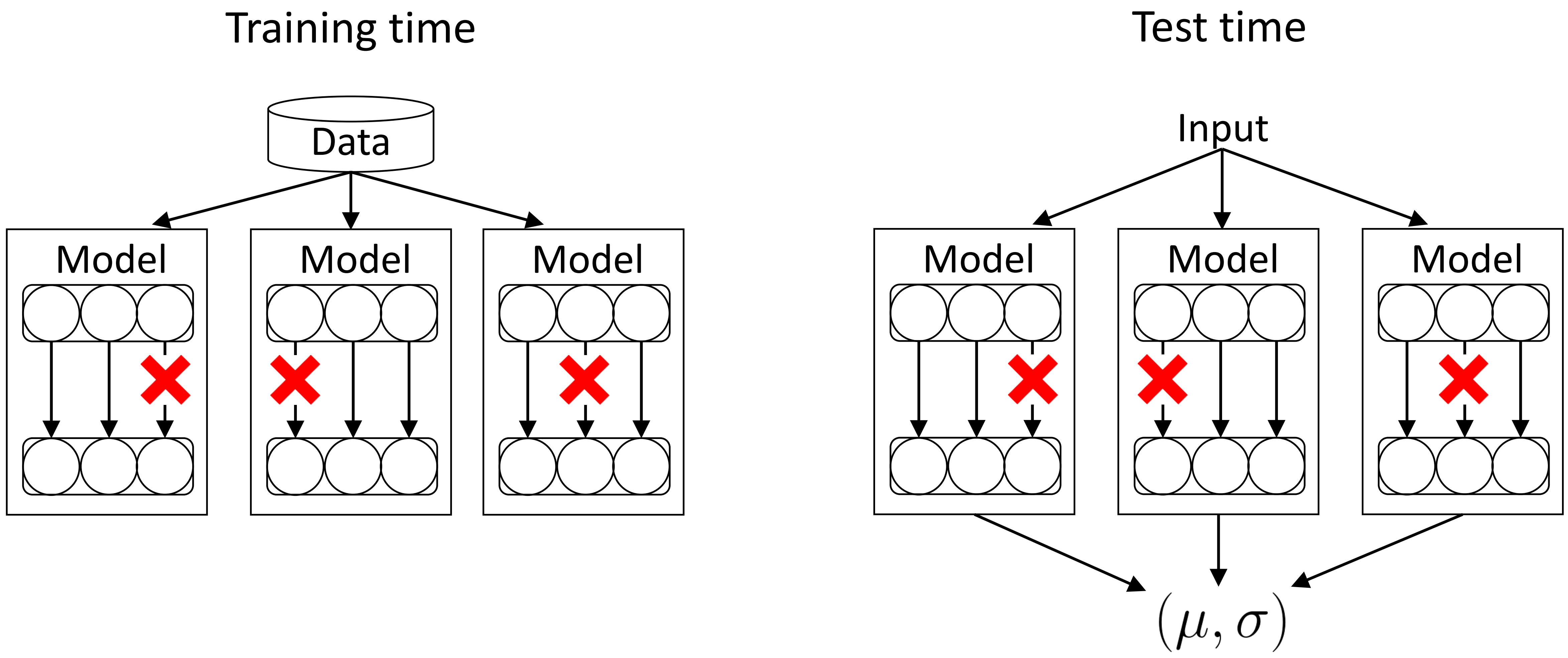


Test time



Estimating neural network output uncertainty

Dropout



Preliminary real-world experiments

Not accounting for uncertainty
(higher-speed collisions)


Speed (m/s)
0.00
0.05
0.10
0.15
0.20
0.25
Time (s)
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Preliminary real-world experiments

accounting for uncertainty
(lower-speed collisions)

Preliminary real-world experiments

successful flight past obstacle

Safety takeaways

- Tradeoff between safety and exploration
- Safety guarantees require expert oversight or known environment + dynamics
- Uncertainty can play a key role

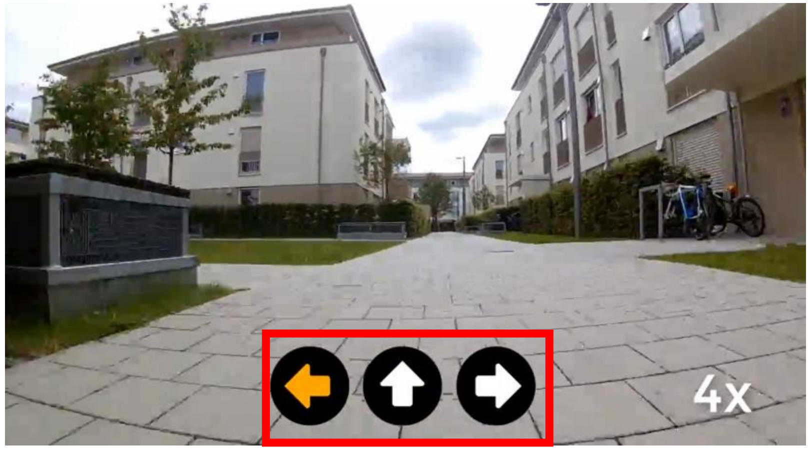
End-to-end Driving via Conditional Imitation Learning

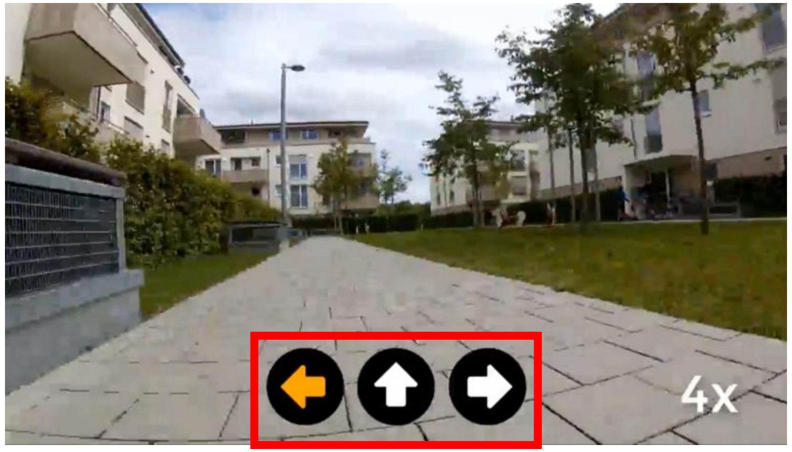
Felipe Codevilla[1,2] Matthias Müller[1,3] Antonio López[2] Vladlen Koltun[1] Alexey Dosovitskiy[1]

Goal

$$\min_{\theta} \|\pi_\theta(\mathbf{a}|\mathbf{s}) - \pi^*(\mathbf{a}|\mathbf{s})\| \quad \longrightarrow \quad \min_{\theta} \|\pi_\theta(\mathbf{a}|\mathbf{s}, \underline{\mathbf{c}}) - \pi^*(\mathbf{a}|\mathbf{s}, \underline{\mathbf{c}})\|$$

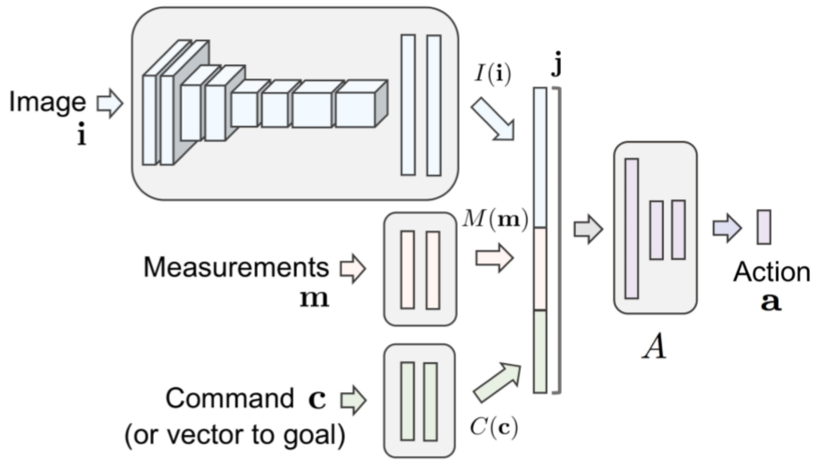
User-specified command



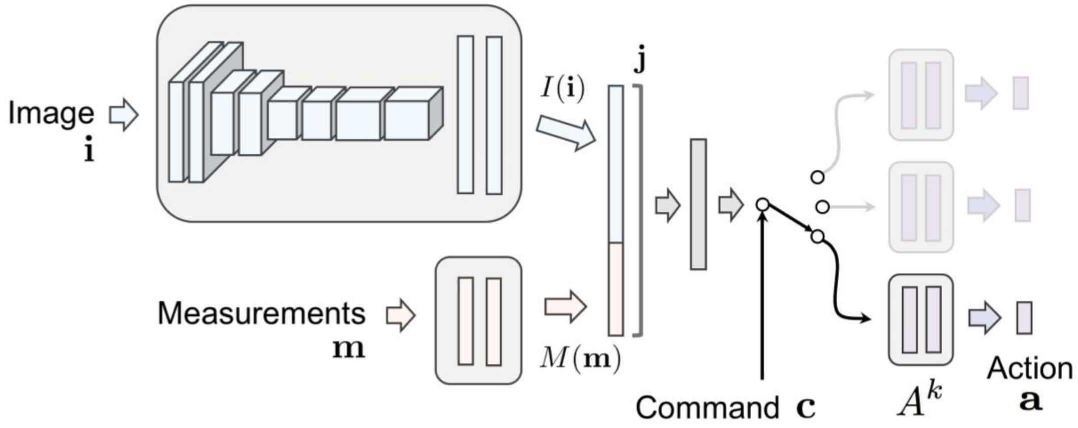


Approach

Option A: Input command



Option B: Branch using command



+ empirically better
- only works for discrete commands

Approach

Important details

- Data augmentation
  - Contrast
  - Brightness
  - Tone
  - Gaussian blur
  - Salt-and-pepper noise
  - Region dropout
- Adding noise to expert


GAS
BRAKE
4X


4x

Generalization

We evaluate how our model generalizes to previously unseen environments with very different appearance.

Composable Deep Reinforcement Learning for Robotic Manipulation

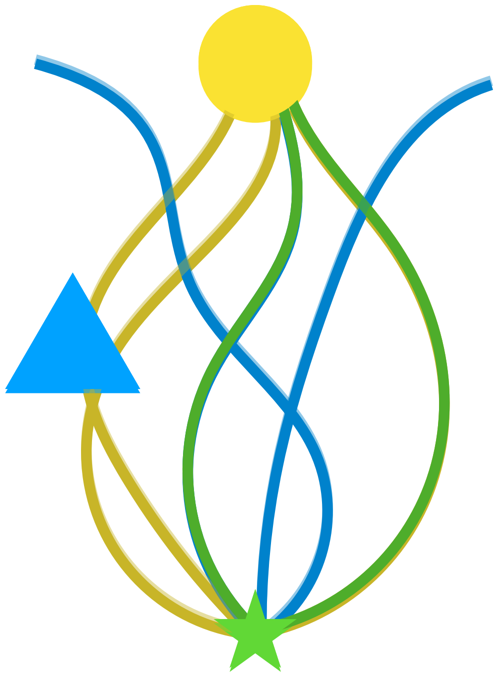
Tuomas Haarnoja[1], Vitchyr Pong[1], Aurick Zhou[1], Murtaza Dalal[1], Pieter Abbeel[1,2], Sergey Levine[1]

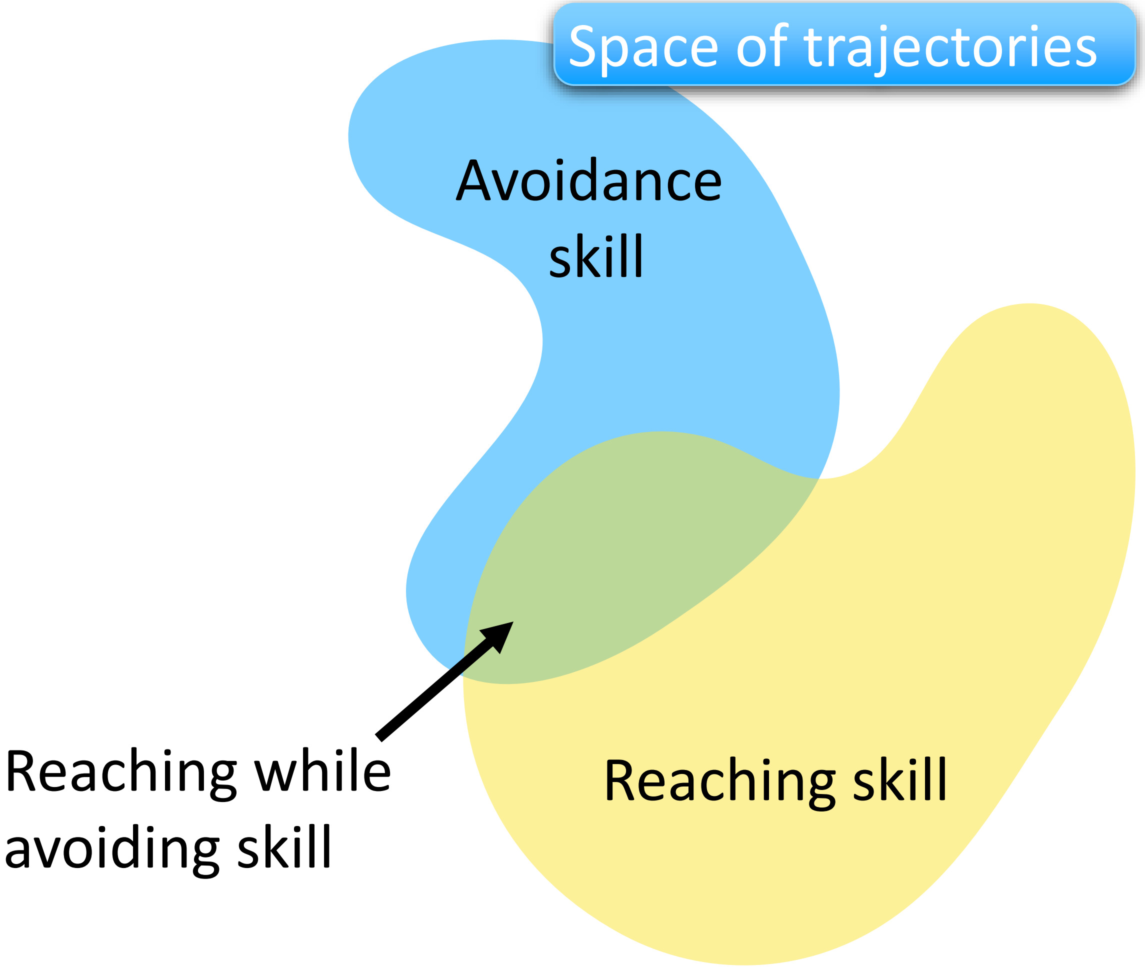
[slides adapted from Tuomas Haarnoja]

Goal

Task 1: Reach

Task 2: Avoid





Policy Composition

Task 1: Reach

$$r_1(\mathbf{s}_t, \mathbf{a}_t)$$
$$Q_1^*(\mathbf{s}_t, \mathbf{a}_t)$$

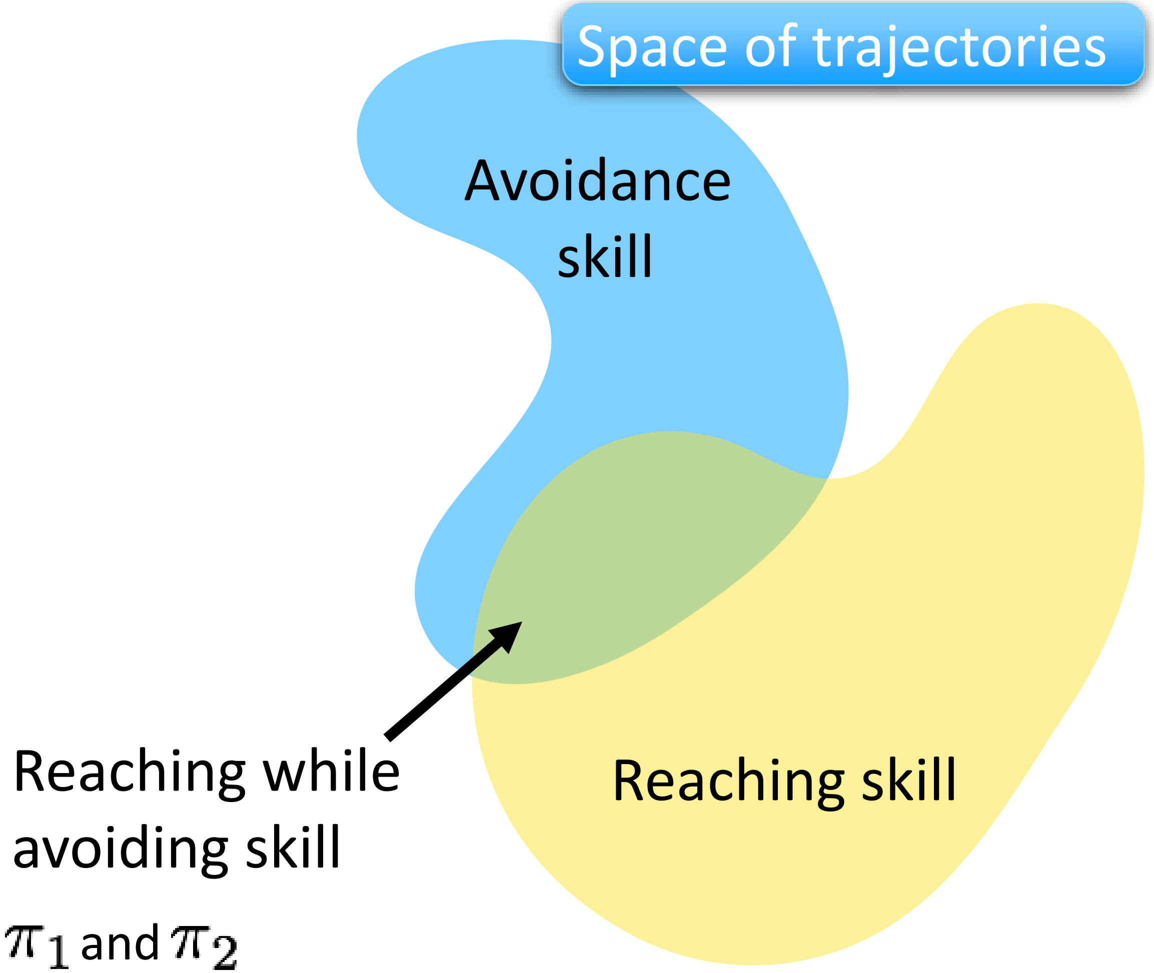
Task 2: Avoid

$$r_2(\mathbf{s}_t, \mathbf{a}_t)$$
$$Q_2^*(\mathbf{s}_t, \mathbf{a}_t)$$

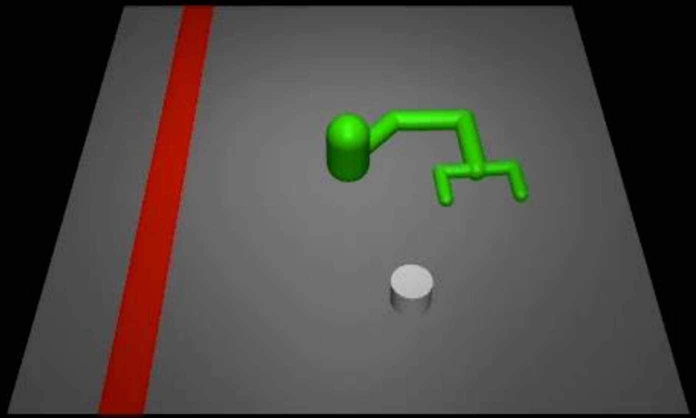
Task 1+2: Reach and avoid

$$r_\mathcal{C} = r_1(\mathbf{s}_t, \mathbf{a}_t) + r_2(\mathbf{s}_t, \mathbf{a}_t)$$
$$Q_\mathcal{C}^*(\mathbf{s}_t, \mathbf{a}_t) \approx Q_1^*(\mathbf{s}_t, \mathbf{a}_t) + Q_2^*(\mathbf{s}_t, \mathbf{a}_t)$$
$$Q_\mathcal{C}^{\pi\Sigma}(\mathbf{s}_t, \mathbf{a}_t) \geq Q_\mathcal{C}^*(\mathbf{s}_t, \mathbf{a}_t) - \boxed{D^*(\mathbf{s}_t, \mathbf{a}_t)}$$

Related to divergence between π_1 and π_2

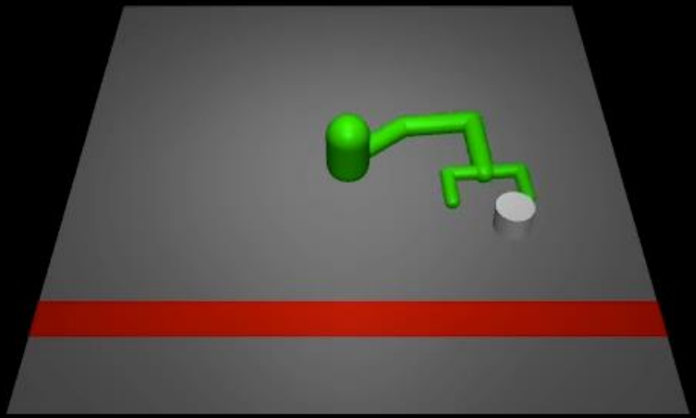


Reusability!



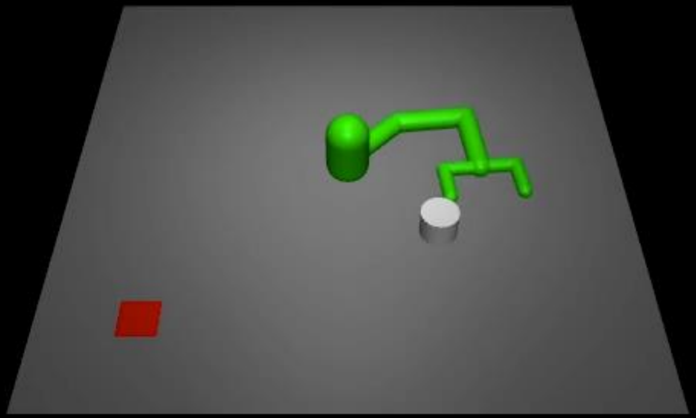
Task 1

$Q_1^*(\mathbf{s}_t, \mathbf{a}_t)$



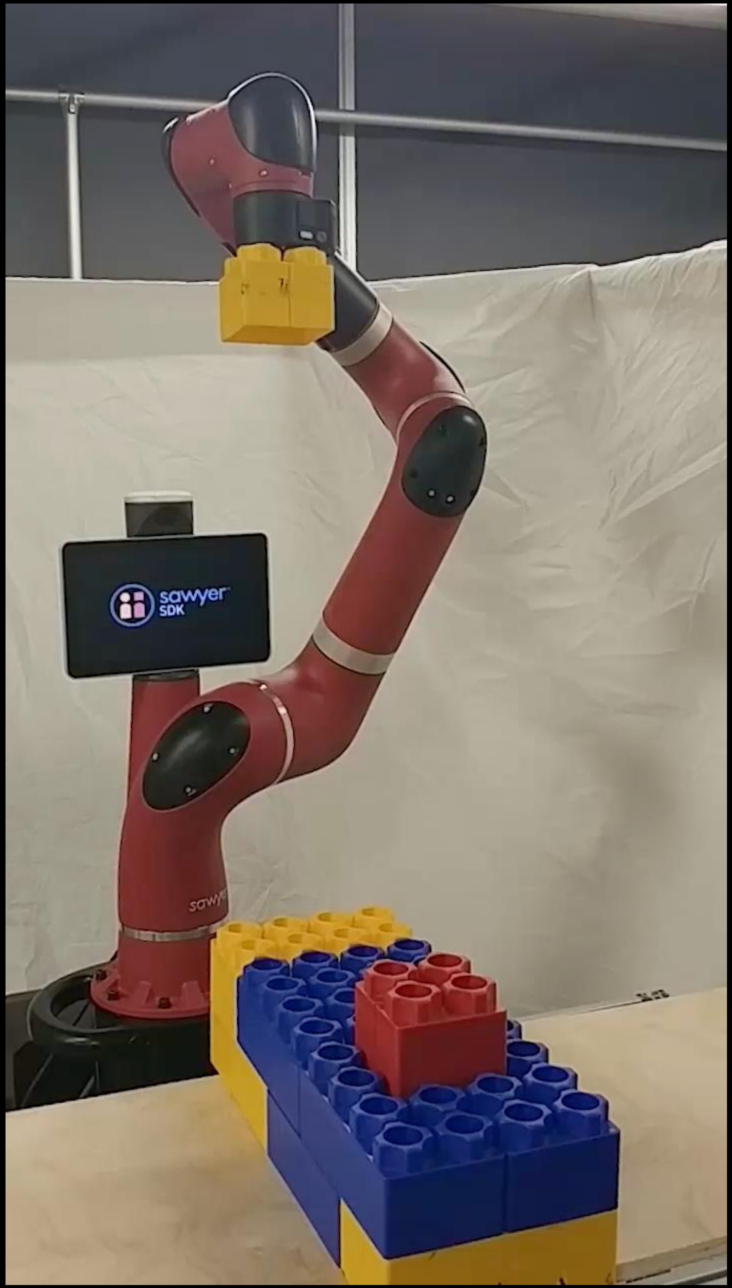
Task 2

$Q_2^*(\mathbf{s}_t, \mathbf{a}_t)$

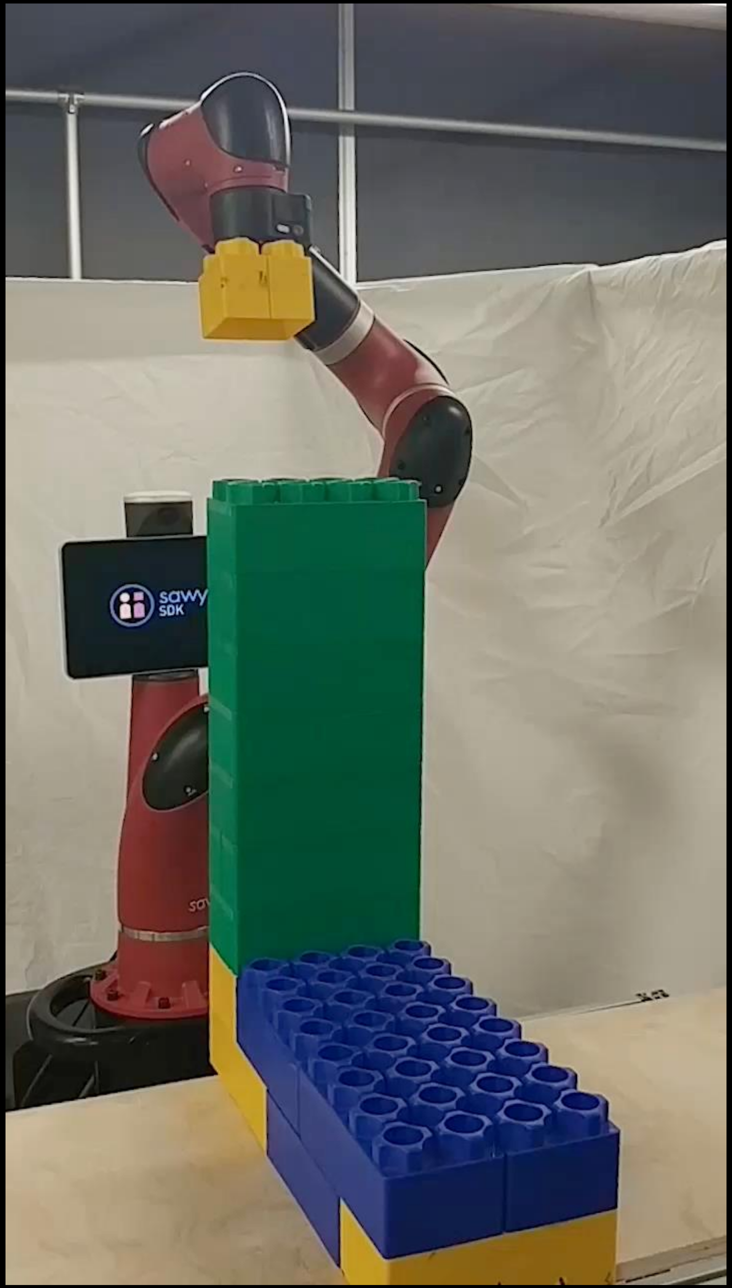


Task 1 + 2

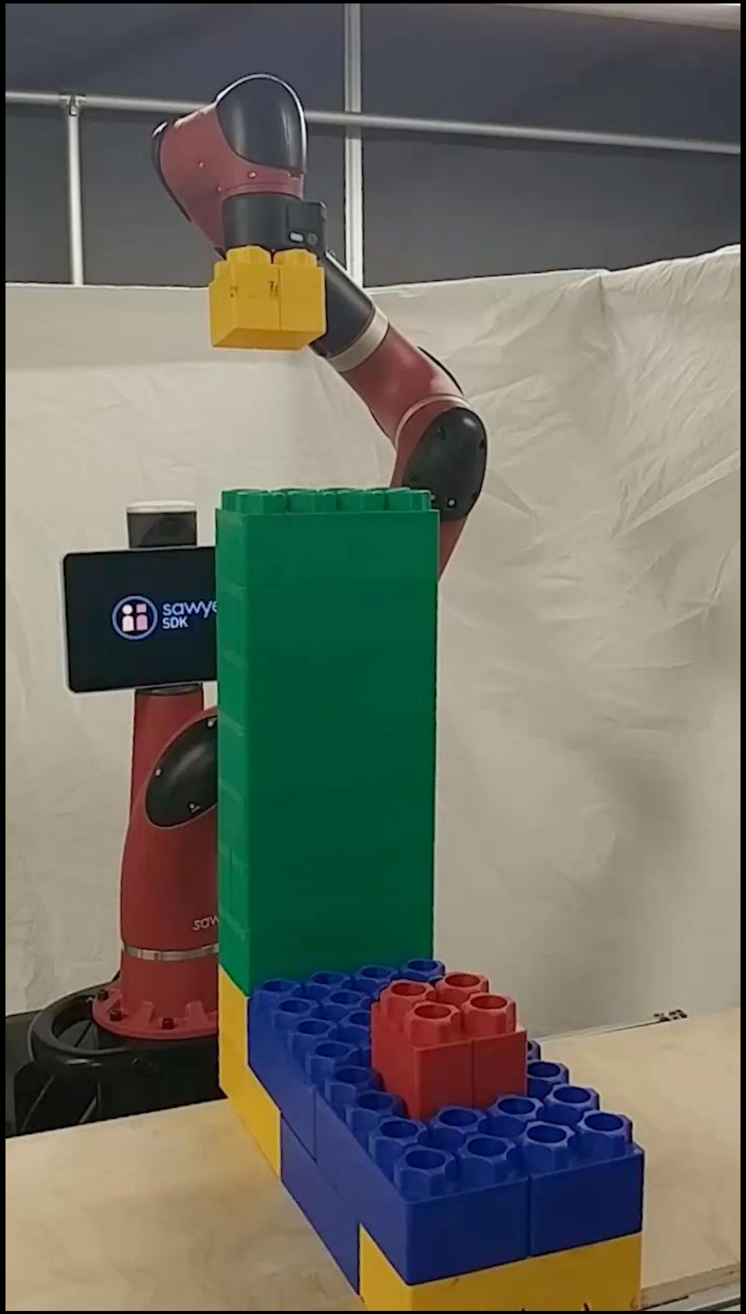
$Q_1^*(\mathbf{s}_t, \mathbf{a}_t) + Q_2^*(\mathbf{s}_t, \mathbf{a}_t)$

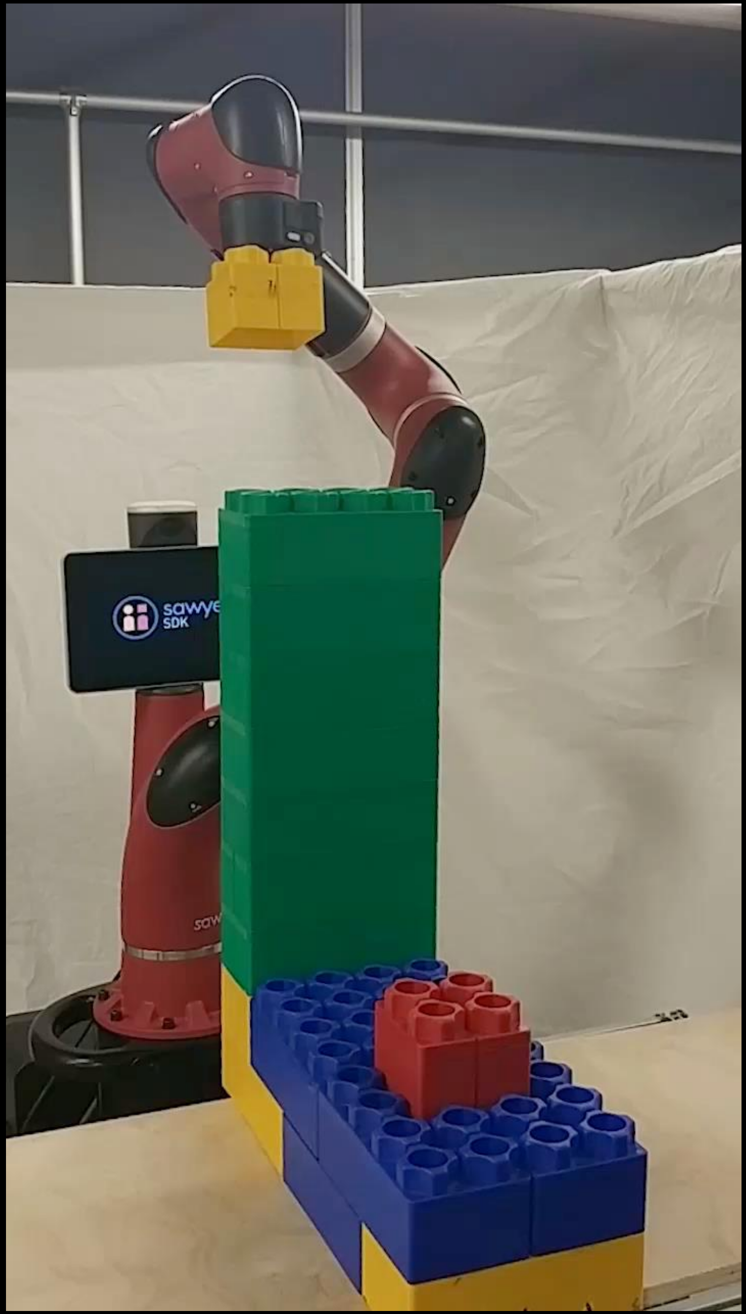


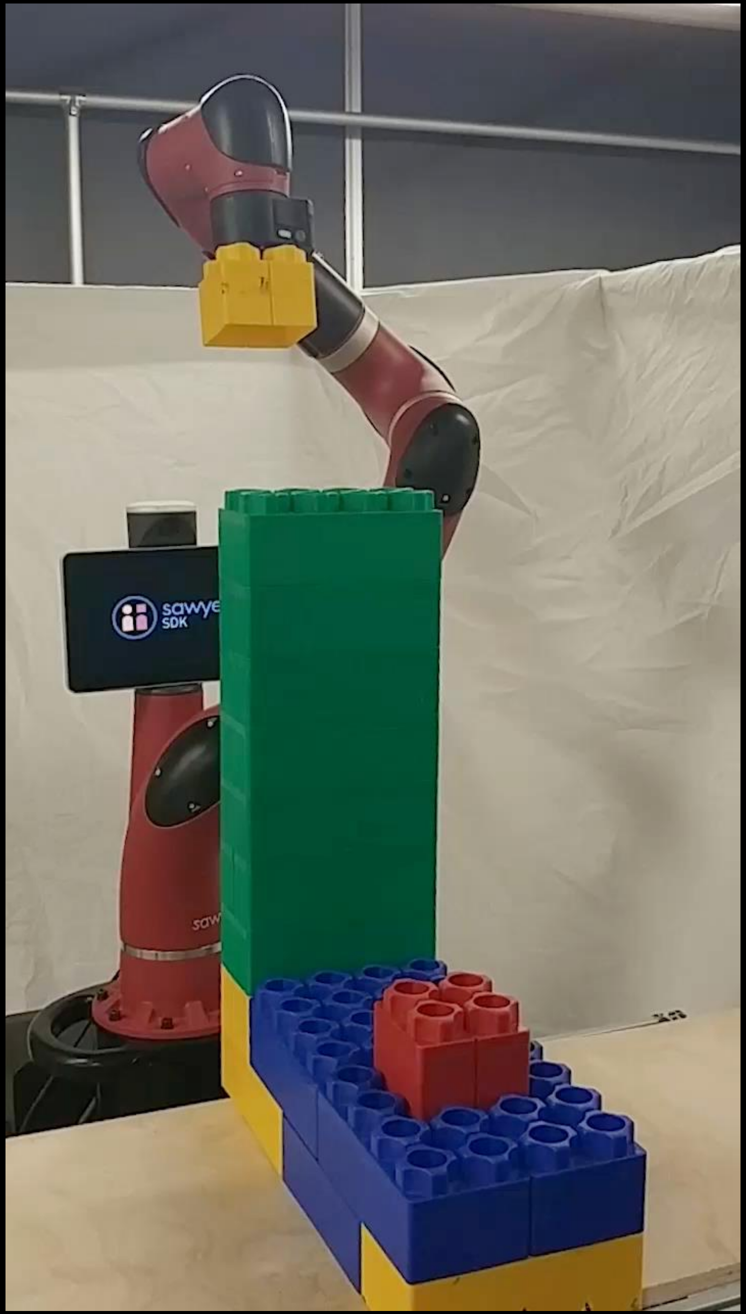
Stacking policy



Avoidance policy


Stacking policy


Avoidance policy


Combined policy

Composable Action-Conditioned Predictors: Flexible Off-Policy Learning for Robot Navigation

Gregory Kahn*, Adam Villaflor*, Pieter Abbeel, Sergey Levine

Standard Reinforcement Learning

Train

Test

Data inefficient

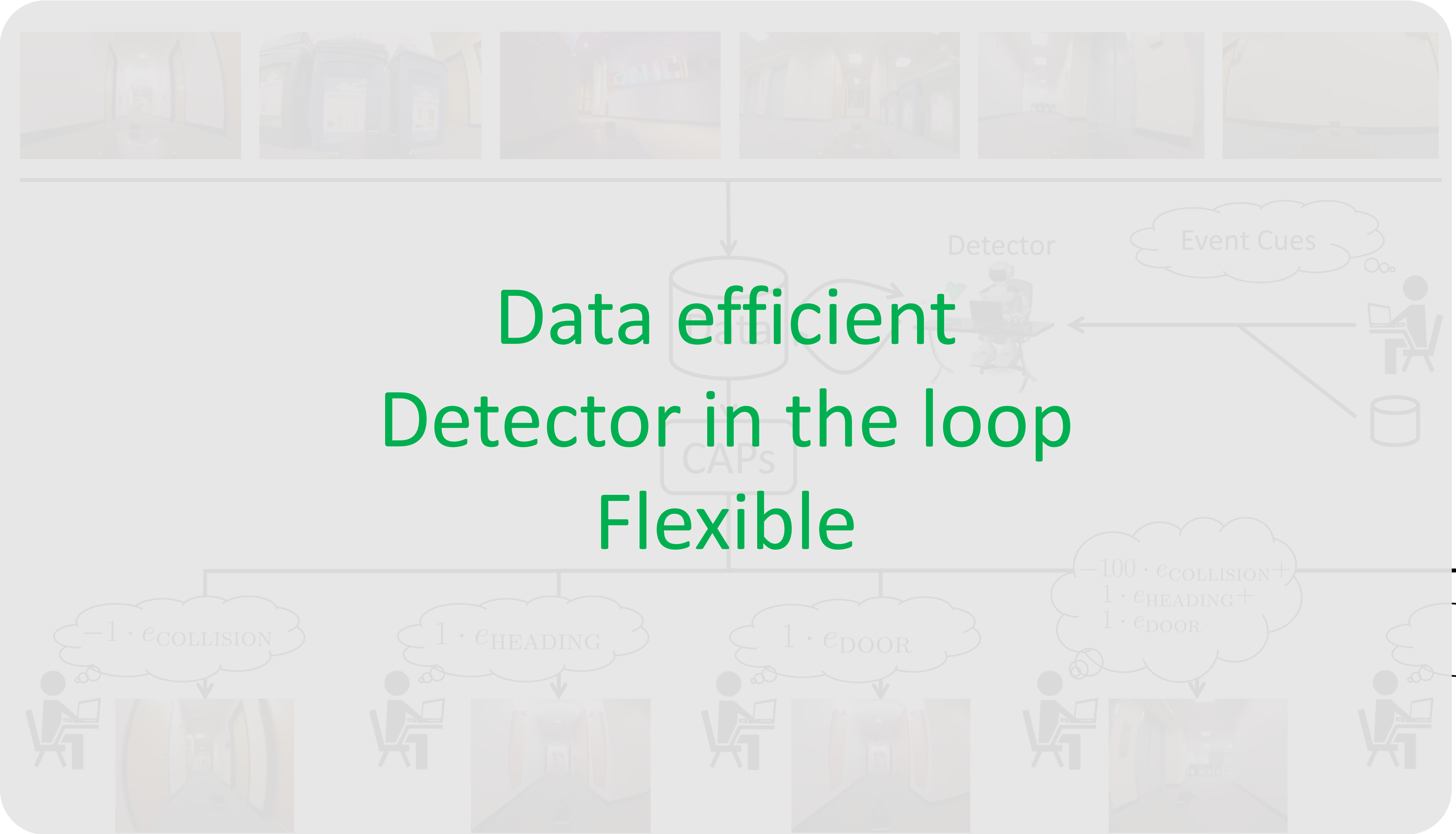
Expert in the loop

Inflexible

CAPs Approach

Train

Test

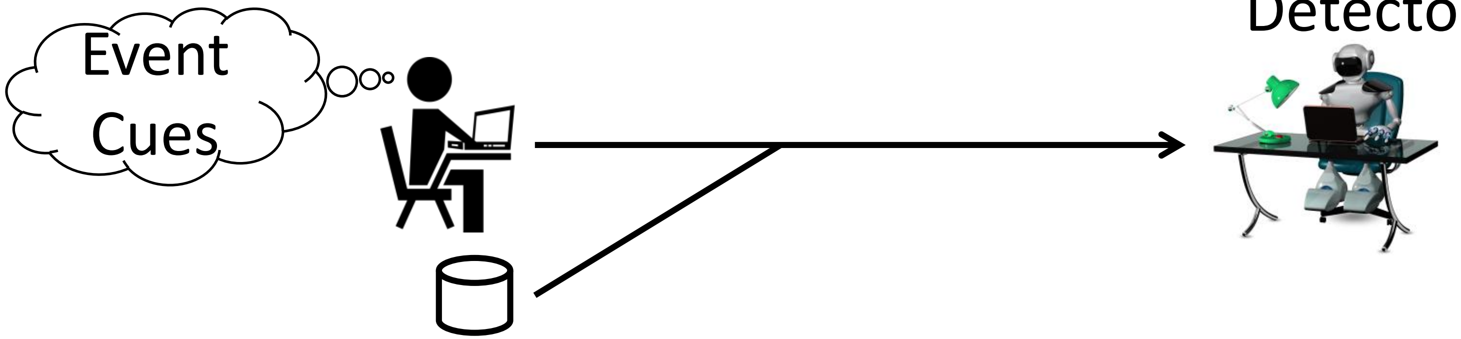


Data efficient

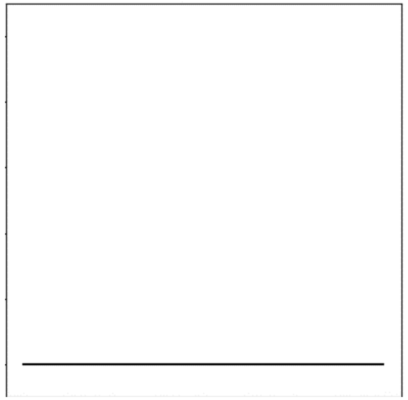
Detector in the loop

Flexible

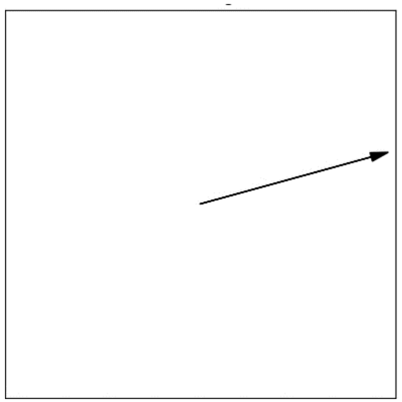
Detect Predict Control



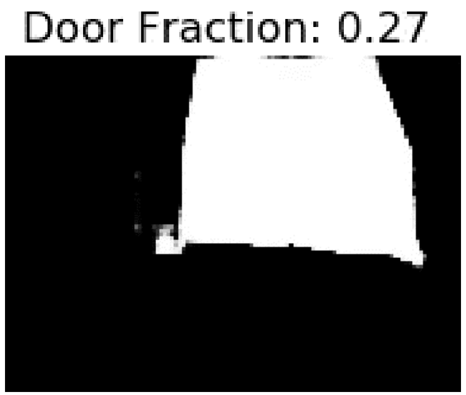
$e_{\text{COLLISION}}$

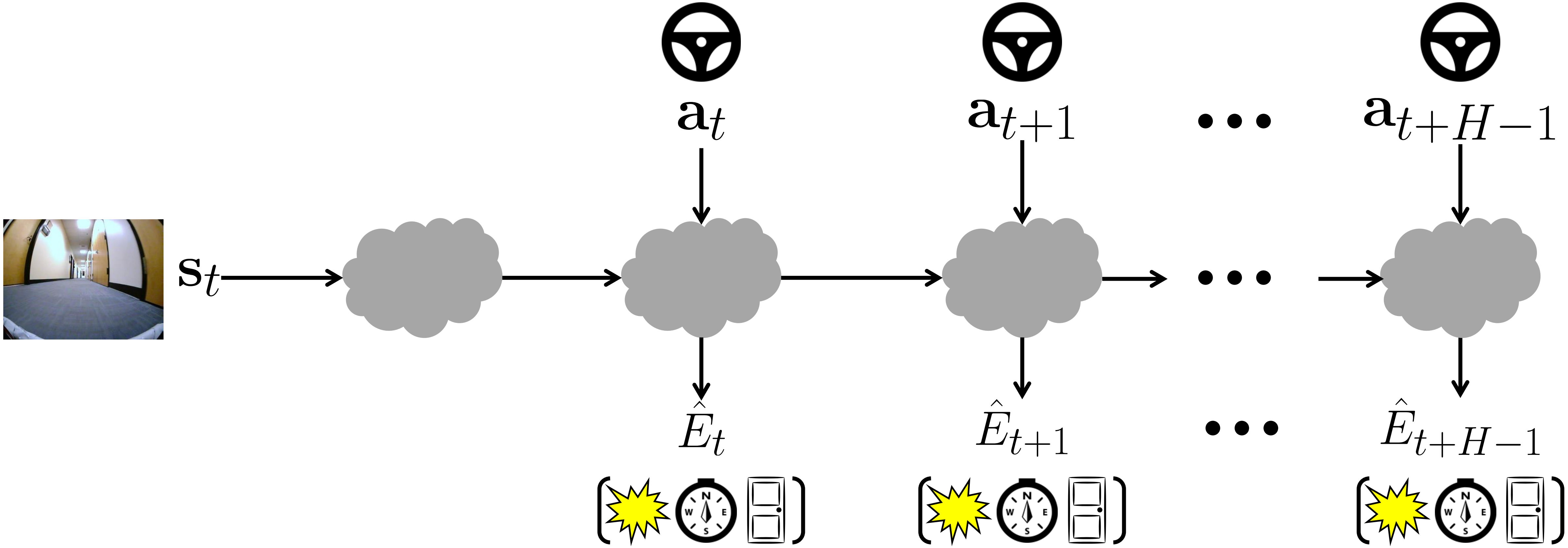


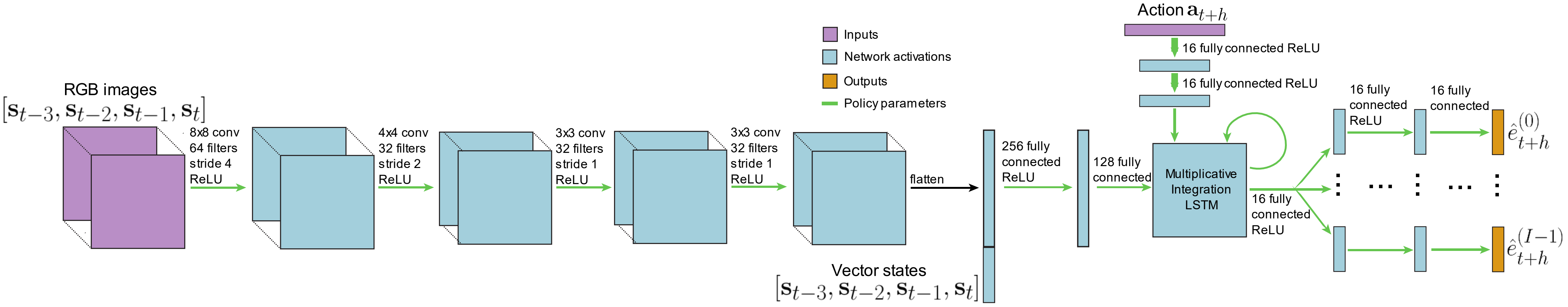
e_{HEADING}

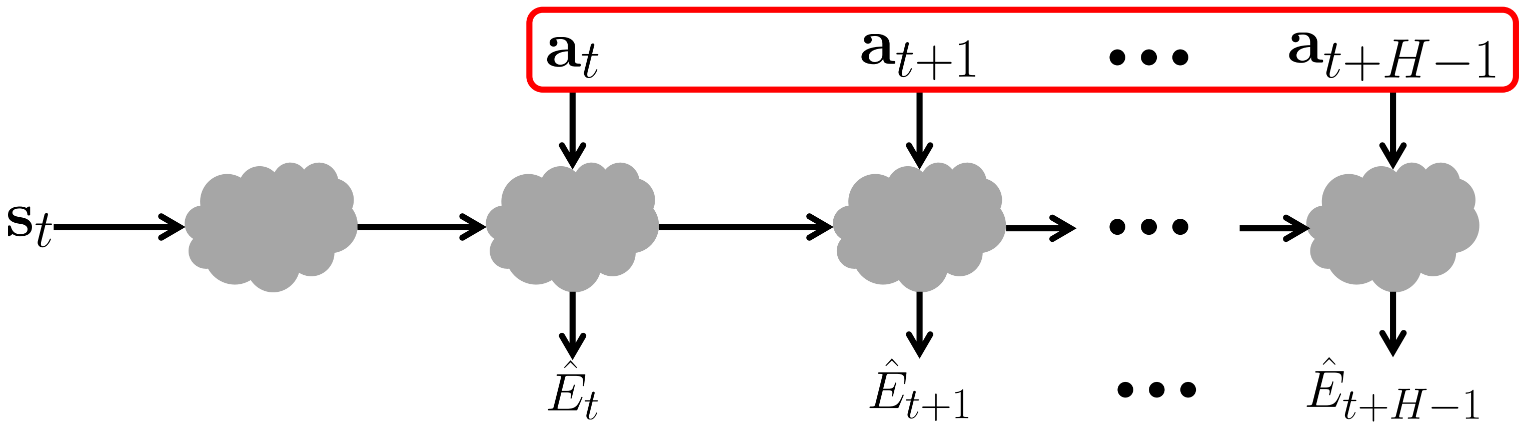


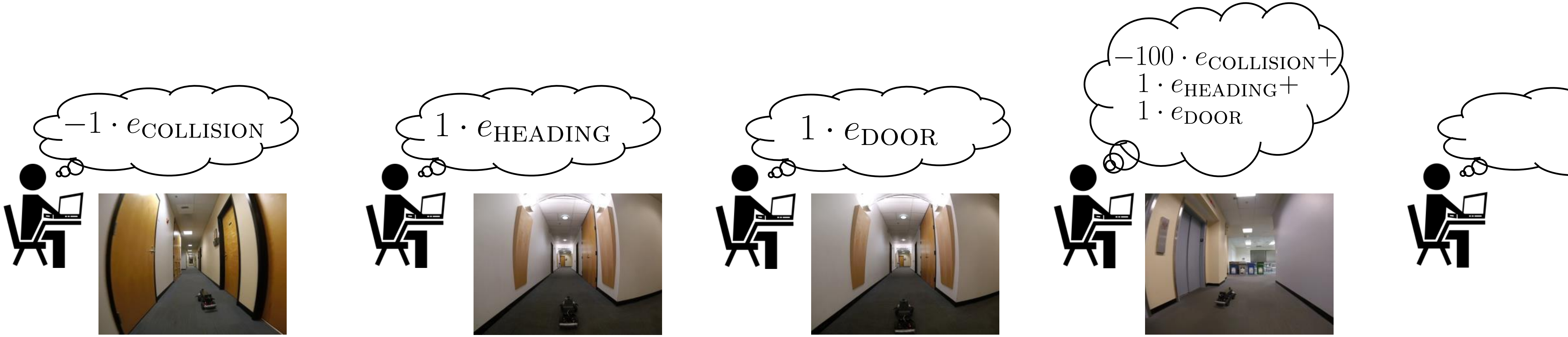
e_{DOOR}




$\mathbf{s}_t$
$\mathbf{a}_t$
$\mathbf{a}_{t+1}$
$\mathbf{a}_{t+H-1}$
$\hat{E}_t$
$\hat{E}_{t+1}$
$\hat{E}_{t+H-1}$
N
W
E
S


RGB images
$[\mathbf{s}_{t-3}, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t]$
8x8 conv
64 filters
stride 4
ReLU
4x4 conv
32 filters
stride 2
ReLU
3x3 conv
32 filters
stride 1
ReLU
3x3 conv
32 filters
stride 1
ReLU
Inputs
Network activations
Outputs
Policy parameters
flatten
Vector states
$[\mathbf{s}_{t-3}, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t]$
256 fully connected ReLU
128 fully connected
Action $\mathbf{a}_{t+h}$
16 fully connected ReLU
16 fully connected ReLU
Multiplicative Integration LSTM
16 fully connected ReLU
16 fully connected ReLU
16 fully connected
$\hat{e}^{(0)}_{t+h}$
$\hat{e}^{(I-1)}_{t+h}$


$\mathbf{a}_t$ $\mathbf{a}_{t+1}$ $\cdots$ $\mathbf{a}_{t+H-1}$
$\mathbf{s}_t$
$\hat{E}_t$ $\hat{E}_{t+1}$ $\cdots$ $\hat{E}_{t+H-1}$


$-1 \cdot e_{\text{COLLISION}}$
$1 \cdot e_{\text{HEADING}}$
$1 \cdot e_{\text{DOOR}}$
$-100 \cdot e_{\text{COLLISION}} + 1 \cdot e_{\text{HEADING}} + 1 \cdot e_{\text{DOOR}}$


8x

8x

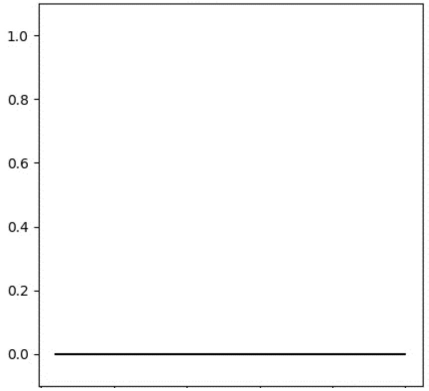
8x

8x

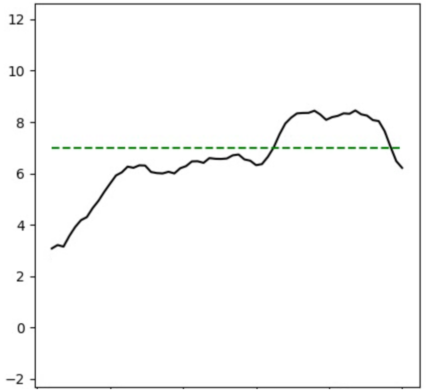
8x

8x

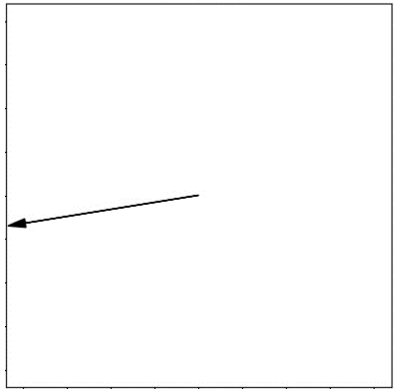
$e_{\text{COLLISION}}$



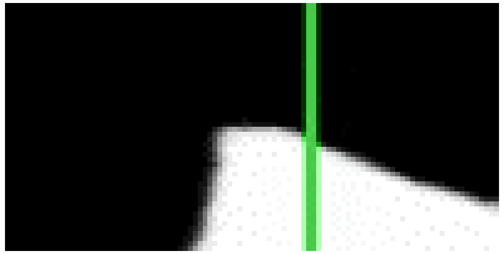
e_{SPEED}



e_{HEADING}



$e_{\text{LANE\_SEEN}}$
$e_{\text{LANE\_DIFF}}$

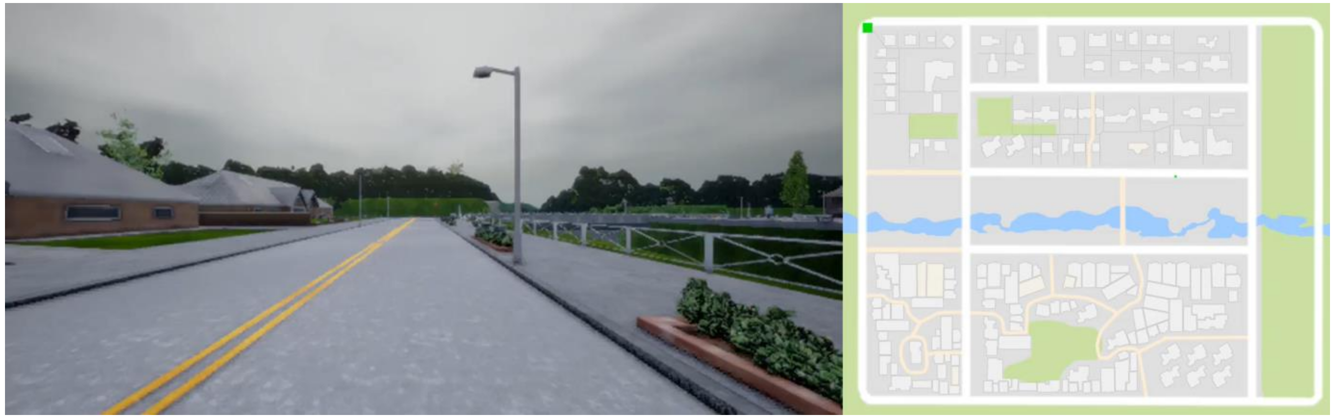


Drive at 7m/s
Avoid collisions

Drive in either lane

Drive in right lane

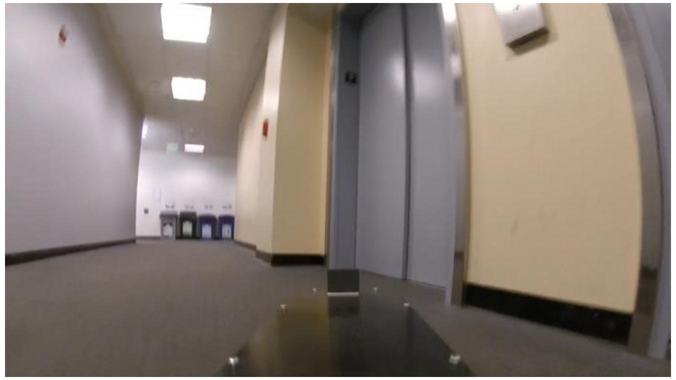
CAPs



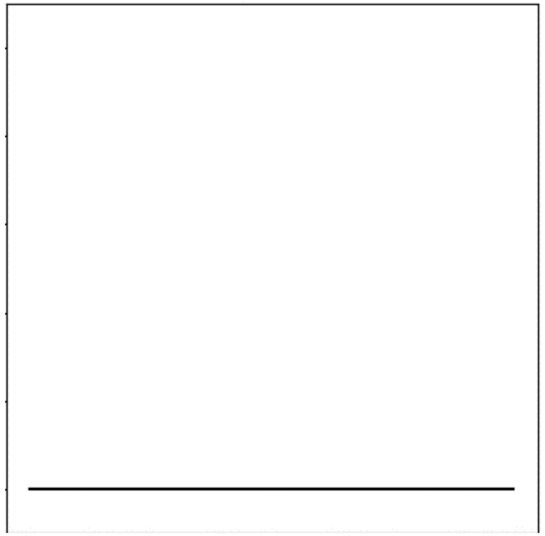



COMPOST
CANS/BOTTLES
MIXED PAPER

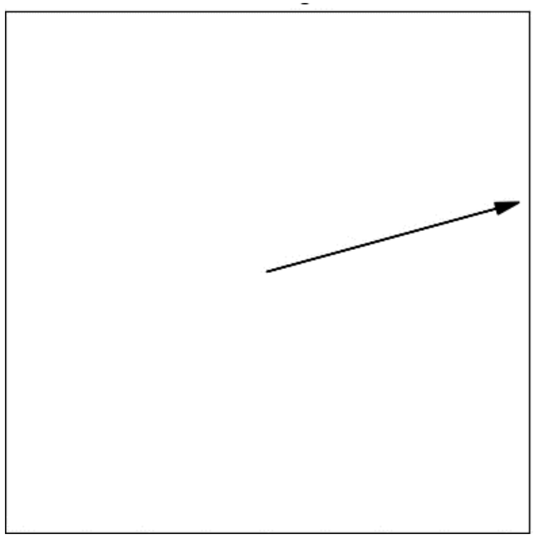

THE DONALD O. PEDERSON
CENTER FOR ELECTRONIC SYSTEMS DESIGN



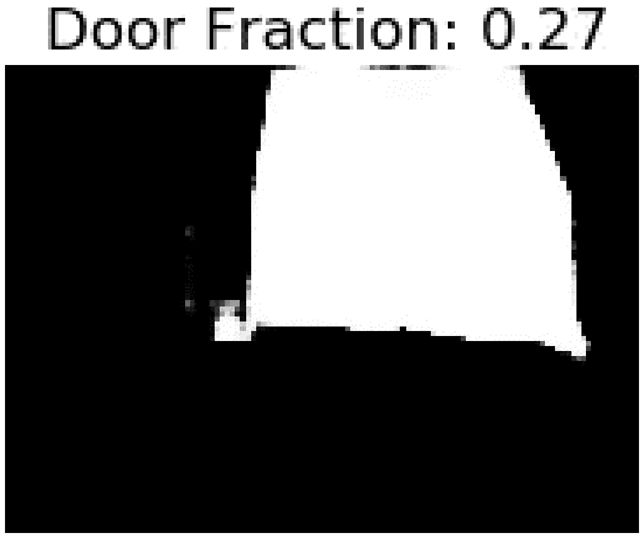
$e_{\text{COLLISION}}$



e_{HEADING}



e_{DOOR}



Collision Avoidance

CAPs

DQL


Avoid collisions
Follow goal heading
Move towards doors
Heading

Flexibility takeaways

- Carefully construct how your policy / model deals with goals
- Model-free methods require extra care to reuse
- Model-based methods are flexible by construction