

Imitation Learning & Behavioral Cloning

Lucas Janson and Sham Kakade

**CS/Stat 184: Introduction to Reinforcement Learning
Fall 2022**

Today

- HW4 will be posted today. (Please start early!)
- Recap++
 - an example + Proximal Policy Optimization (PPO)
- Today:
 1. Overview of PG, problems+successes
 2. Behavior Cloning

Recap++

Some Helpful Notation: Visitation Measures

- Visitation probability at time h : $\mathbb{P}_h(s_h, a_h \mid \mu, \pi)$
- Average Visitation Measure:

$$d_\mu^\pi(s, a) = \frac{1}{H} \sum_{h=0}^{H-1} \mathbb{P}_h(s, a \mid \mu, \pi)$$

- With this def, we have:

$$\begin{aligned} J(\theta) &:= E_{s_0 \sim \mu_0} [V^{\pi_\theta}(s_0)] \\ &= E \left[\sum_{h=0}^{H-1} r(s_h, a_h) \mid \mu_0, \pi_\theta \right] \\ &= H \cdot \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} E_{a \sim \pi_\theta(s)} [r(s, a)] \end{aligned}$$

TRPO

At iteration t , with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\theta} H \cdot \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta}(s)} A^{\pi_{\theta_t}}(s, a) \right]$$

$$\text{s.t.}, KL \left(\rho_{\pi_{\theta_t}} \mid \rho_{\pi_{\theta}} \right) \leq \delta$$

We want to maximize local advantage against π_{θ_t} , but we want the new policy to be close to π_{θ_t} (in the KL sense)

How we can actually do the optimization here?
After all, we don't even know the analytical form of trajectory likelihood...

NPG derived from TRPO:

We did second-order Taylor expansion on the KL constraint, and we get:

$$\frac{1}{H} KL \left(\rho_{\pi_{\theta_t}} | \rho_{\pi_{\theta}} \right) \approx \frac{1}{2} (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t)$$

$$F_{\theta_t} := \mathbb{E}_{s, a \sim d_{\mu}^{\pi_{\theta_t}}} \left[\nabla_{\theta} \ln \pi_{\theta_t}(a | s) \left(\nabla_{\theta} \ln \pi_{\theta_t}(a | s) \right)^\top \right] \in \mathbb{R}^{dim_{\theta} \times dim_{\theta}}$$

This leads to the following simplified constrained optimization:

$$\begin{aligned} & \max_{\theta} \nabla_{\theta} J(\pi_{\theta_t})^\top (\theta - \theta_t) \\ & \text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta \end{aligned}$$

Algorithm: Natural Policy Gradient

Initialize θ_0

For $t = 0, \dots$

Estimate PG $\nabla_{\theta} J(\pi_{\theta_t})$

Estimate Fisher info-matrix $F_{\theta_t} := \mathbb{E}_{s,a \sim d_{\mu}^{\pi_{\theta_t}}} \nabla_{\theta} \ln \pi_{\theta_t}(a | s) (\nabla_{\theta} \ln \pi_{\theta_t}(a | s))^{\top}$

Natural Gradient Ascent: $\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_{\theta} J(\pi_{\theta_t})$

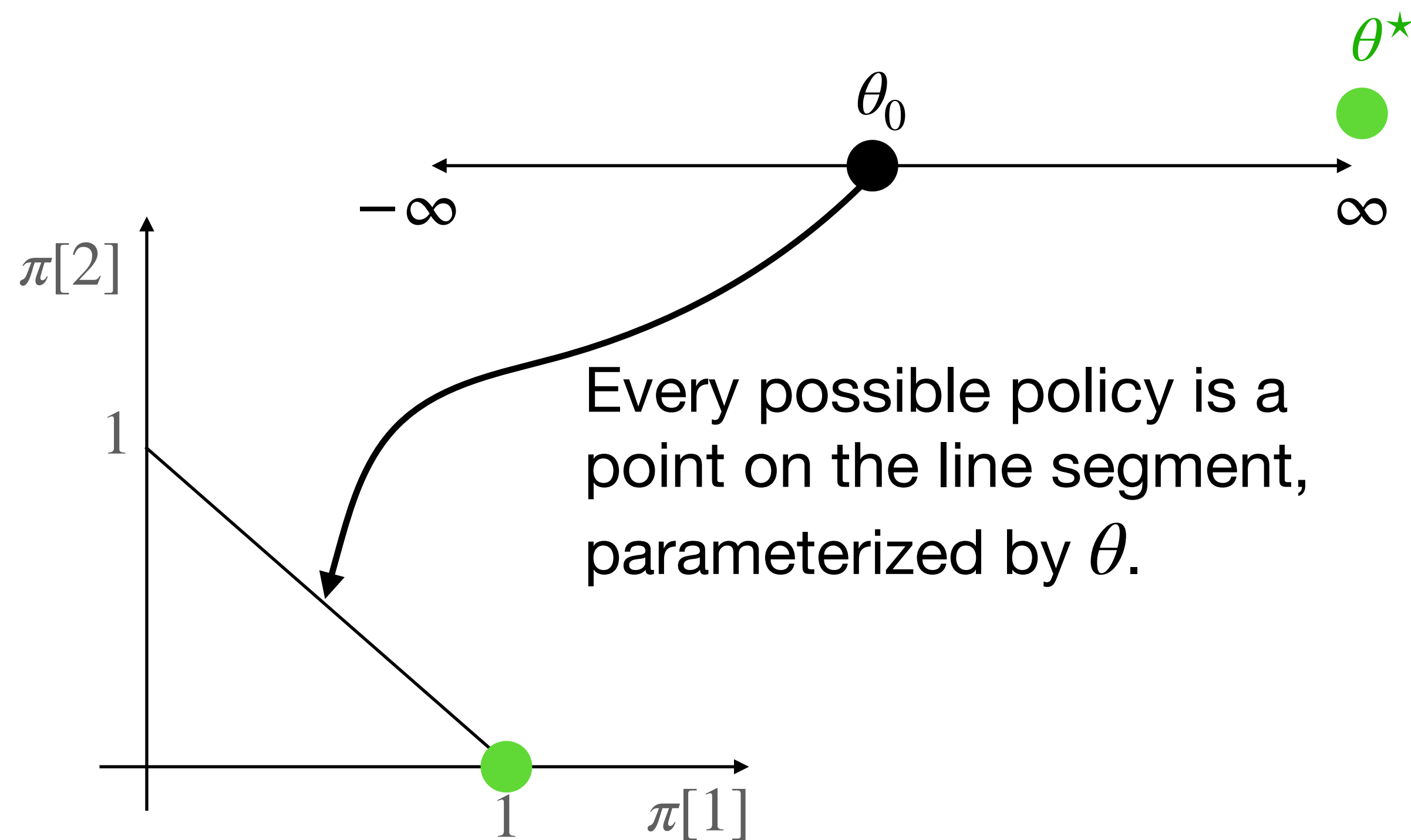
Using a tuned η or using $\eta = \sqrt{\frac{\delta}{\nabla_{\theta} J(\pi_{\theta_t})^{\top} F_{\theta_t}^{-1} \nabla_{\theta} J(\pi_{\theta_t})}}$

(We will implement it in HW4 on Cartpole)

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



$$\text{Gradient: } J'(\theta) = \frac{99 \exp(\theta)}{(1 + \exp(\theta))^2}$$

$$\text{Exact PG: } \theta_{t+1} = \theta_t + \eta \frac{99 \exp(\theta_t)}{(1 + \exp(\theta_t))^2}$$

i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \rightarrow 0$ as $\theta \rightarrow \infty$

$$\text{Fisher information scalar: } f_\theta = \frac{\exp(\theta)}{(1 + \exp(\theta))^2}$$

$$\text{NPG: } \theta_{t+1} = \theta_t + \eta \frac{J'(\theta_t)}{f_{\theta_t}} = \theta_t + \eta \cdot 99$$

NPG moves to $\theta = \infty$ much more quickly (for a fixed η)

Proximal Policy Optimization (PPO): A computationally fast extension of NPG:

Given an current policy π^t , we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta}(\cdot | s)} A^{\pi_{\theta_t}}(s, a) \right] - \underbrace{\lambda \mathbb{E}_{s \sim d_{\mu}^{\pi^t}} \left[\text{KL} \left(\pi_{\theta_t}(a | s) | \pi_{\theta}(a | s) \right) \right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta_t}(\cdot | s)} \frac{\pi_{\theta}(a | s)}{\pi_{\theta_t}(a | s)} A^{\pi_{\theta_t}}(s, a) \right] - \lambda \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \mathbb{E}_{a \sim \pi_{\theta_t}(\cdot | s)} \left[-\ln \pi_{\theta}(a | s) \right]$$

PPO: Perform a few steps of mini-batch SGA on $\ell(\theta)$ to approximate $\arg \max_{\theta} \ell(\theta)$

Today:

Optimality in Markov Decision Processes

Outline:

1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

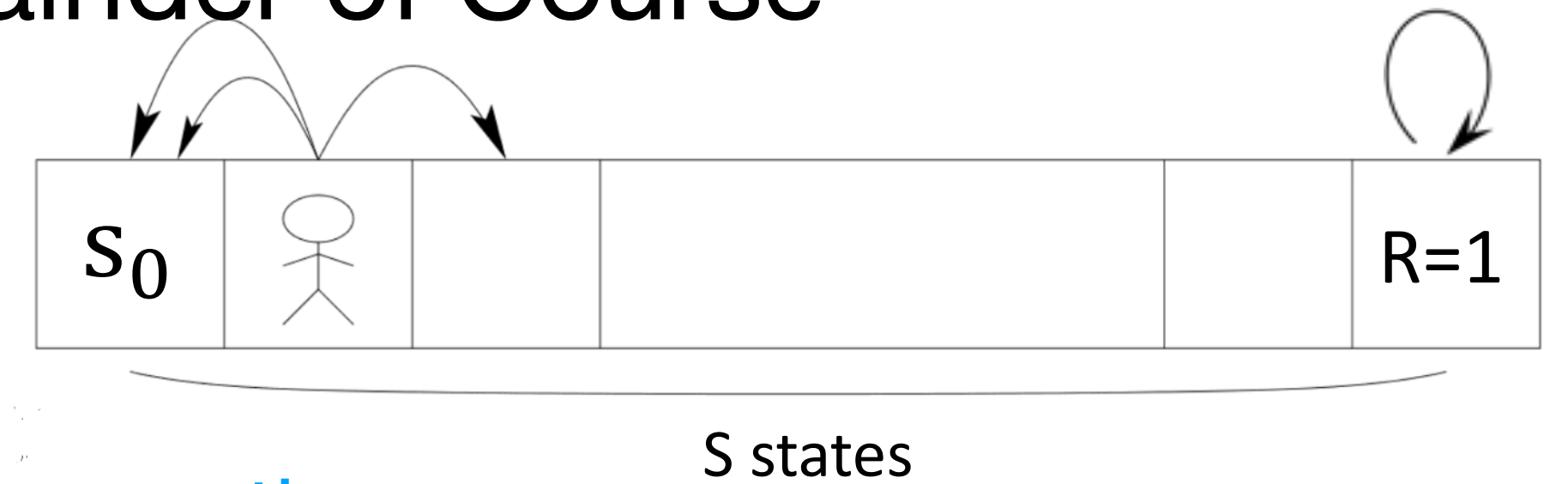
“Lack of Exploration” leads to Optimization and Statistical Challenges



Thrun '92

- Suppose $|S| \approx H$ or $|S| \approx 1/(1 - \gamma)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).
- A randomly initialized policy has prob. $O(1/3^{|S|})$ of hitting the goal state in a single trajectory.
- Implications:
 - Any sample based policy iteration approach (starting with this policy) requires $O(3^{|S|})$ trajectories to make progress at the very first step.
 - Same for any sample based PG method.
 - Related: even if we had exact gradients, the “landscape” is such that these gradients are exponentially small, at randomly initialized policy (see [AJKS Ch 11](#)).

Implications/Comments/Remainder of Course



Thrun '92

- Sometimes exploration is (or can be made) “easier” in practice
 - Random strategies can reach “rewarding milestones”
 - We can design/“shape” the reward function to help us out.
- We can try to make the distribution μ to have better coverage.
 - For small problems, μ being uniform would make all these issues go away. (for large problems, μ being uniform may not help at all. Why?)
 - Ideally, μ having support on where a good policy tends to visit is helpful (sometimes we can't design μ)
- Course:
 - A little theory with regards to μ and PG. (today)
PG has better guarantees than approx DP methods (in terms of μ).
 - Imitation learning (starting today).
An expert gives us samples from a “good” μ .
 - Explicit Exploration: for the “tabular case” (we will mix UCB with VI!)

Outline:

1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

Let's compare fitted DP and PG for “Linear” Parameterizations of Q-functions and Policies

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and
parameter $\theta \in \mathbb{R}^d$

1. Linear Functions

$$f_{\theta}(s, a) = \theta^{\top} \phi(s, a)$$

1. Softmax linear Policy

$$\pi_{\theta}(a | s) = \frac{\exp(\theta^{\top} \phi(s, a))}{\sum_{a'} \exp(\theta^{\top} \phi(s, a'))}$$

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting “state-action” distribution. $J(\pi) = E_{s_0, a_0 \sim \mu}[Q^\pi(s, a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π ,

$$\min_{\theta} E_{s, a \sim \mu} \left[\left(Q^\pi(s, a) - \theta^\top \phi(s, a) \right)^2 \right] \leq \delta, \text{ and } \min_{\theta} \|Q^\pi - \theta^\top \phi\|_\infty \leq \delta_\infty$$
 - δ : the average case supervised learning error (reasonable to expect this can be made small)
 - δ_∞ : the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

- Suppose that we use a # samples that is poly in d & $1/\epsilon_{stat}$ for both fittedPI and NPG.
- FittedPI will return a policy π^{FPI} with the performance guarantee:

$$J(\pi^{FPI}) \geq J(\pi^\star) - \epsilon_{stat} - 2H^2\delta_\infty$$
- NPG has the same guarantee.
- NPG also has a stronger guarantee: Suppose μ has “reasonable support” on where π^\star tends to visit, i.e. suppose:

$$\max_{s, a} \left(\frac{d_\mu^{\pi^\star}(s, a)}{\mu(s, a)} \right) \leq C$$

then NPG will return a policy with sub-optimality determined by C and the average case error δ :

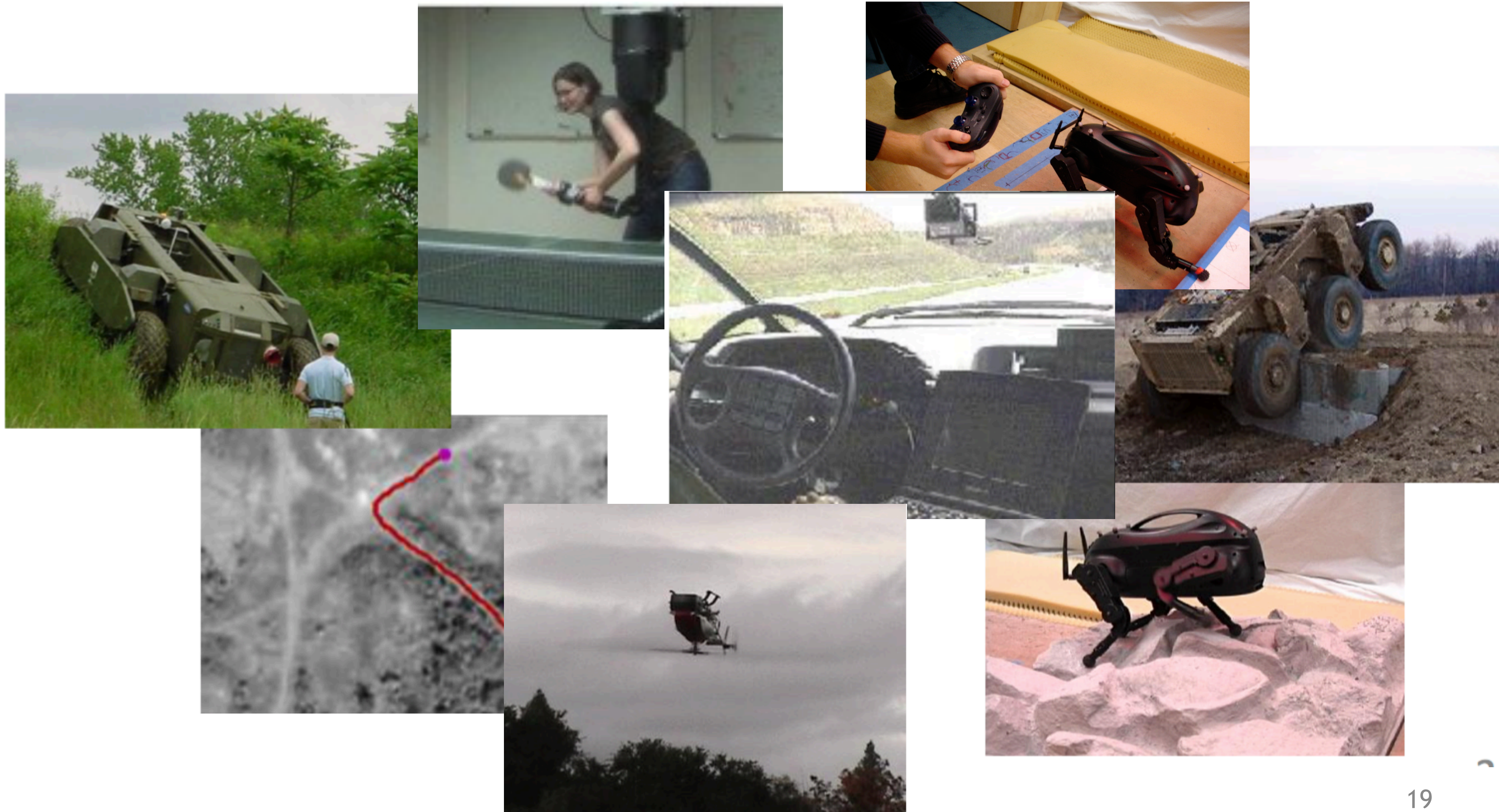
$$J(\pi^{NPG}) \geq J(\pi^\star) - \epsilon_{stat} - 2H^2 C \delta$$

Outline:

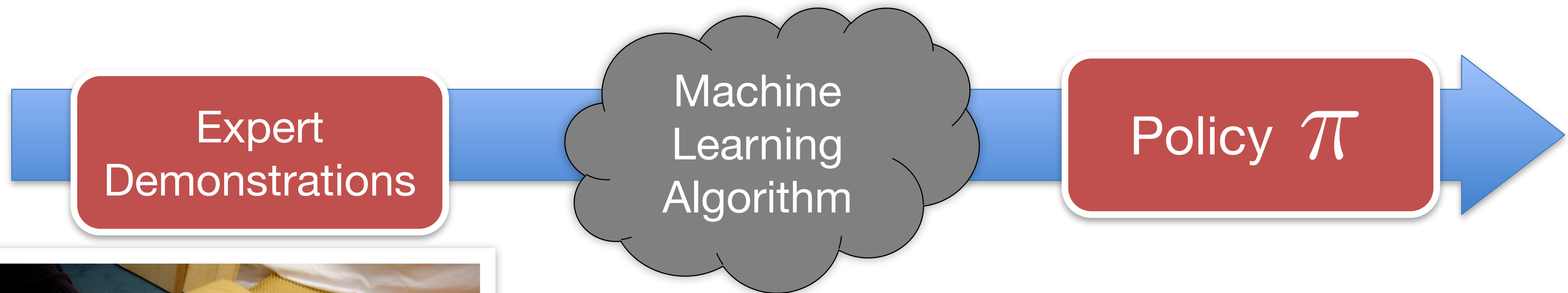
1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

3a. Introduction of Imitation Learning

Imitation Learning



Imitation Learning



- SVM
- Gaussian Process
- Kernel Estimator
- Deep Networks
- Random Forests
- LWR
- ...

Maps *states* to actions

Learning to Drive by Imitation

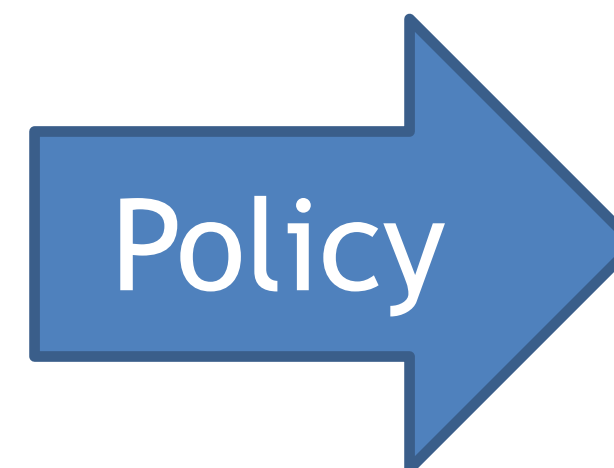
[Pomerleau89, Saxena05, Ross11a]

Input:



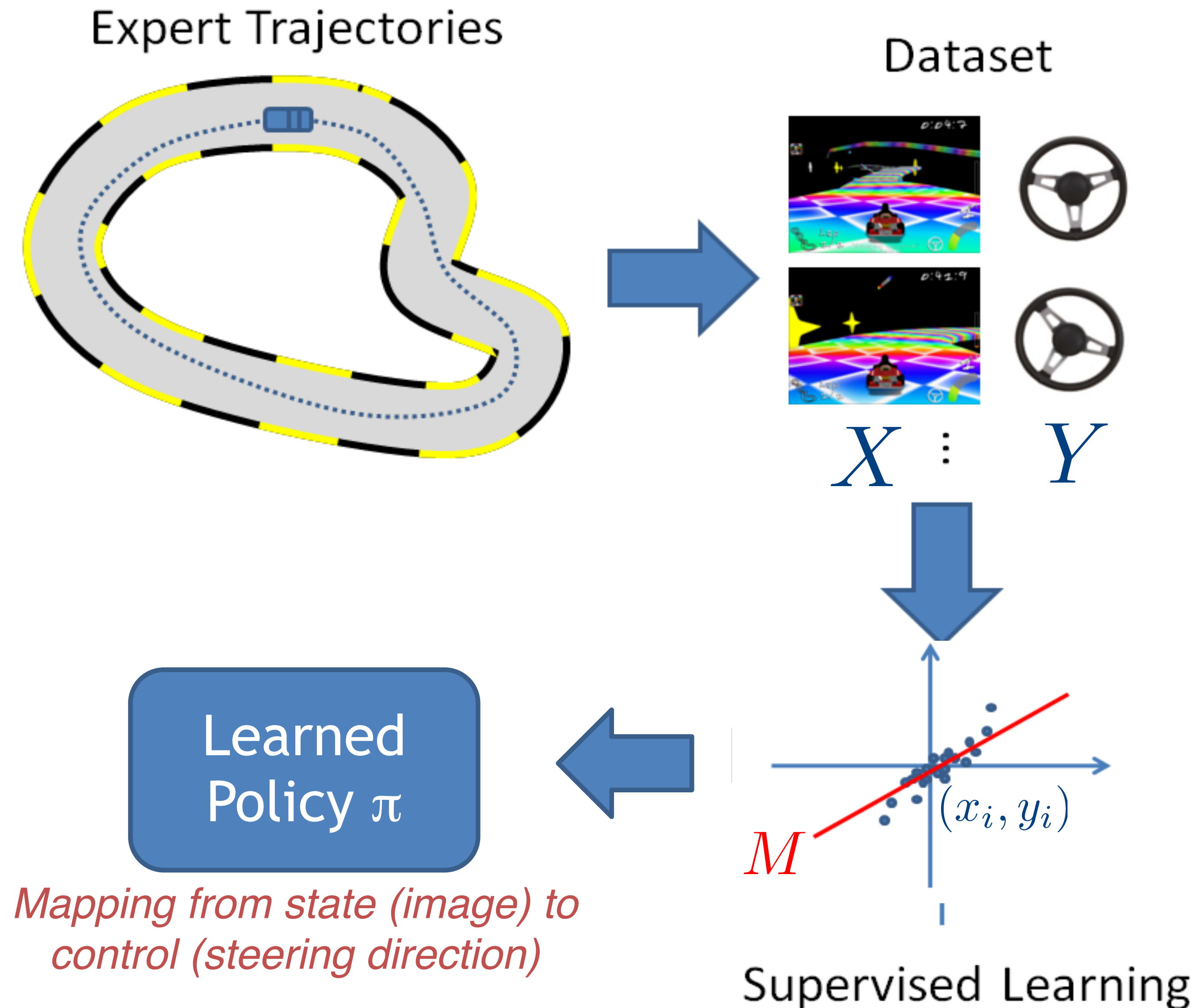
Camera Image

Output:



Steering Angle
in $[-1, 1]$

Supervised Learning Approach: Behavior Cloning



3b. Offline Imitation Learning: Behavior Cloning

Let's formalize the offline IL Setting and the Behavior Cloning algorithm

Discounted infinite horizon MDP $\mathcal{M} = \{S, A, \gamma, r, P, \rho, \pi^\star\}$

Ground truth reward $r(s, a) \in [0, 1]$ is unknown;
For simplicity, let's assume expert is a (nearly) optimal policy π^\star

We have a dataset $\mathcal{D} = (s_i^\star, a_i^\star)_{i=1}^M \sim d^{\pi^\star}$

Goal: learn a policy from \mathcal{D} that is as good as the expert π^\star

Let's formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class $\Pi = \{\pi : S \mapsto \Delta(A)\}$

BC is a Reduction to Supervised Learning:

$$\hat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^M \ell(\pi, s^{\star}, a^{\star})$$

Many choices of loss functions:

1. Negative log-likelihood (NLL): $\ell(\pi, s, a^{\star}) = -\ln \pi(a^{\star} | s)$
2. square loss (i.e., regression for continuous action): $\ell(\pi, s, a^{\star}) = \|\pi(s) - a^{\star}\|_2^2$

$$\hat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^M \ell(\pi, s^{\star}, a^{\star})$$

Analysis

Assumption: we are going to assume Supervised Learning succeeded

$$\mathbb{E}_{s \sim d_{\mu}^{\pi^{\star}}} \mathbf{1} \left[\hat{\pi}(s) \neq \pi^{\star}(s) \right] \leq \epsilon \in \mathbb{R}^+$$

Note that here training and testing mismatch at this stage!

Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$

The quadratic amplification is annoying

Summary:

1. TRPO/NPG/PPO
2. Exploration/ μ /Guarantees
3. Behavioral Cloning

1-minute feedback form: <https://bit.ly/3RHtlxy>

