Imitation Learning & Behavioral Cloning

Lucas Janson

CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Today

- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

Feedback from feedback forms

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Today

- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

All Policy Gradient Algorithms in One Slide

Parameterize policy and optimize directly while sampling from MDP

Policy Gradient (PG)

Variance too high

Variance reduction techniques like mini-batches and baselines

Fitted Policy Iteration

Big steps unstable

Trust Region Policy Optimization (TRPO)

Approximation in closed form

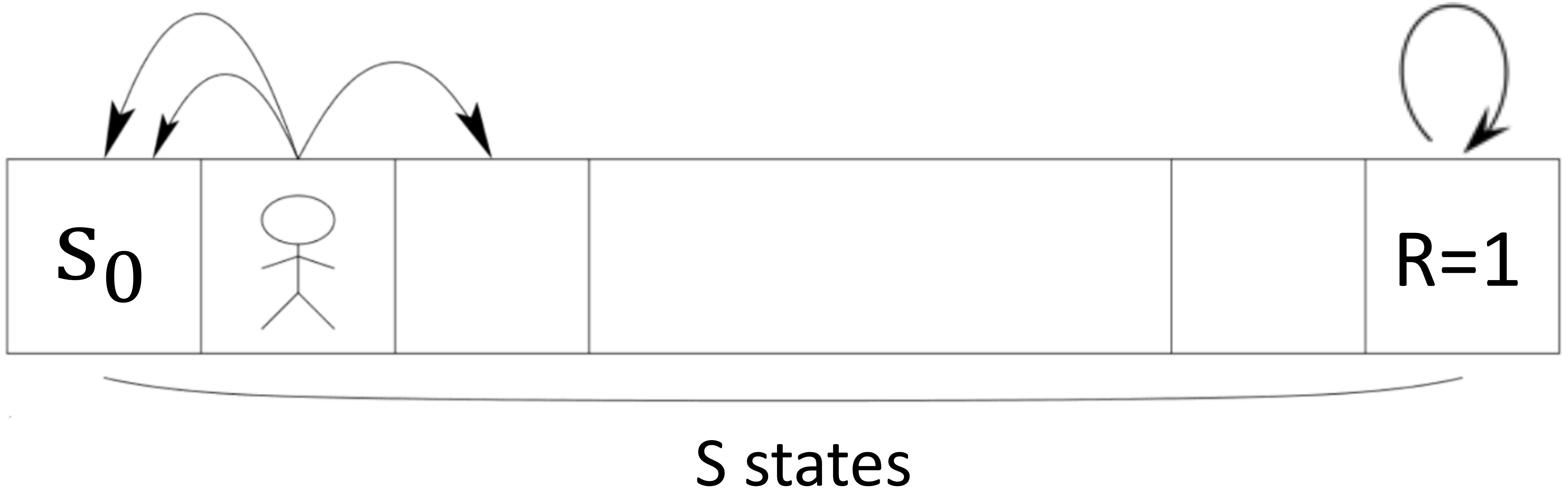
Natural Policy Gradient (NPG)

2nd-order updates expensive

Proximal Policy Optimization (PPO)

PPO gets 2nd-order optimization benefits over PG and 1st-order computation benefits over TRPO/NPG

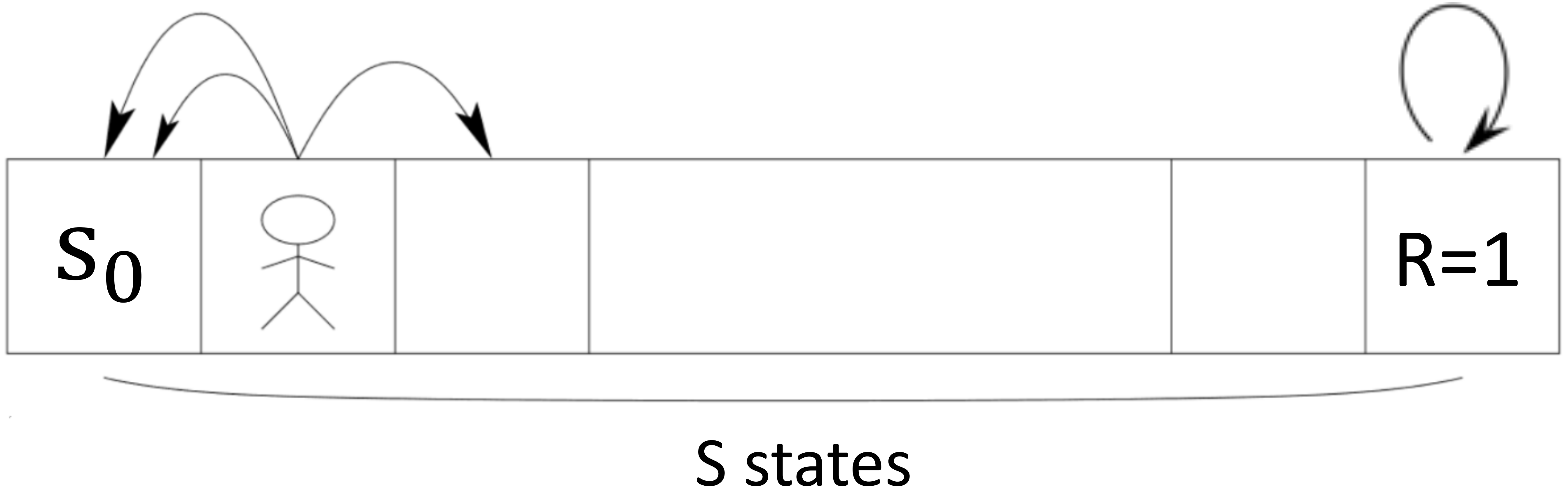
"Lack of Exploration" leads to Optimization and Statistical Challenges



Thrun '92

- Suppose $H \approx \text{poly}(|S|)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).
- A randomly initialized policy π^0 has prob. $O(1/3^{|S|})$ of hitting the goal state in a trajectory.
- Thus a sample-based approach, with $\mu(s_0) = 1$, require $O(3^{|S|})$ trajectories.
    - Holds for (sample based) Fitted DP
    - Holds for (sample based) PG/TRPO/NPG/PPO
- Basically, for these approaches, there is no hope of learning the optimal policy if $\mu(s_0) = 1$.

"Lack of Exploration" leads to Optimization and Statistical Challenges

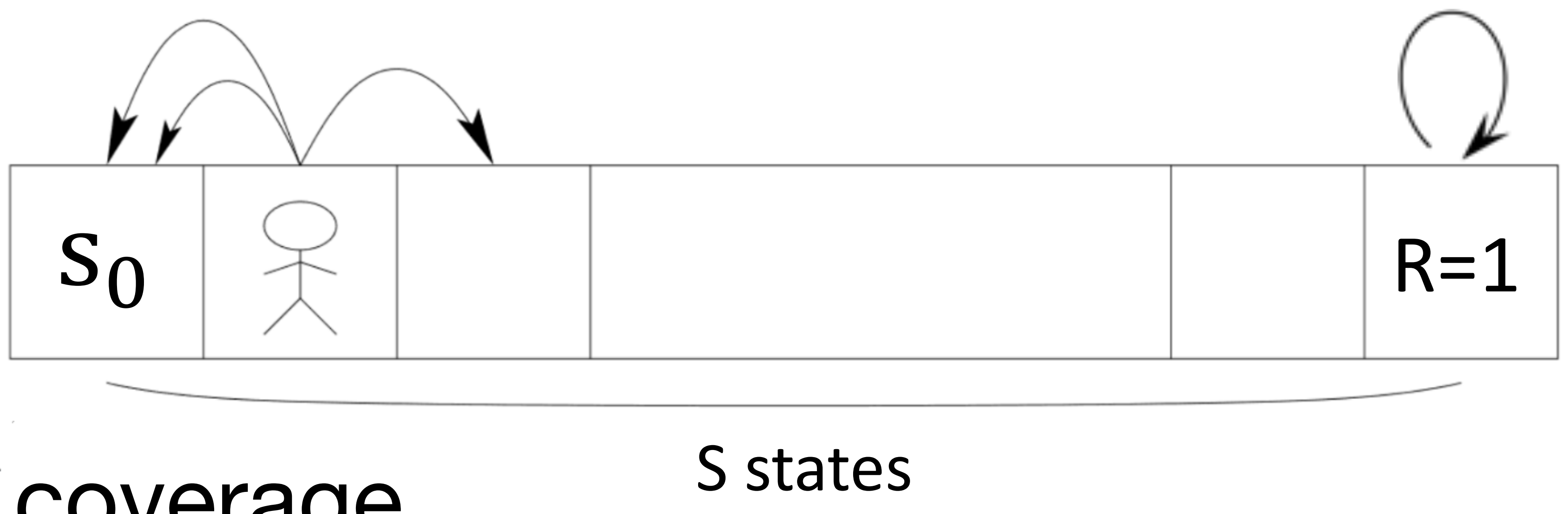


Thrun '92

- Suppose $H \approx \text{poly}(|S|)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).
- A randomly initialized policy π^0 has prob. $O(1/3^{|S|})$ of hitting the goal state in a trajectory.
- Thus a sample-based approach, with $\mu(s_0) = 1,$ require $O(3^{|S|})$ trajectories.
  - Holds for (sample based) Fitted DP
  - Holds for (sample based) PG/TRPO/NPG/PPO
- Basically, for these approaches, there is no hope of learning the optimal policy if $\mu(s_0) = 1.$

Why not do one trajectory that always moves right?

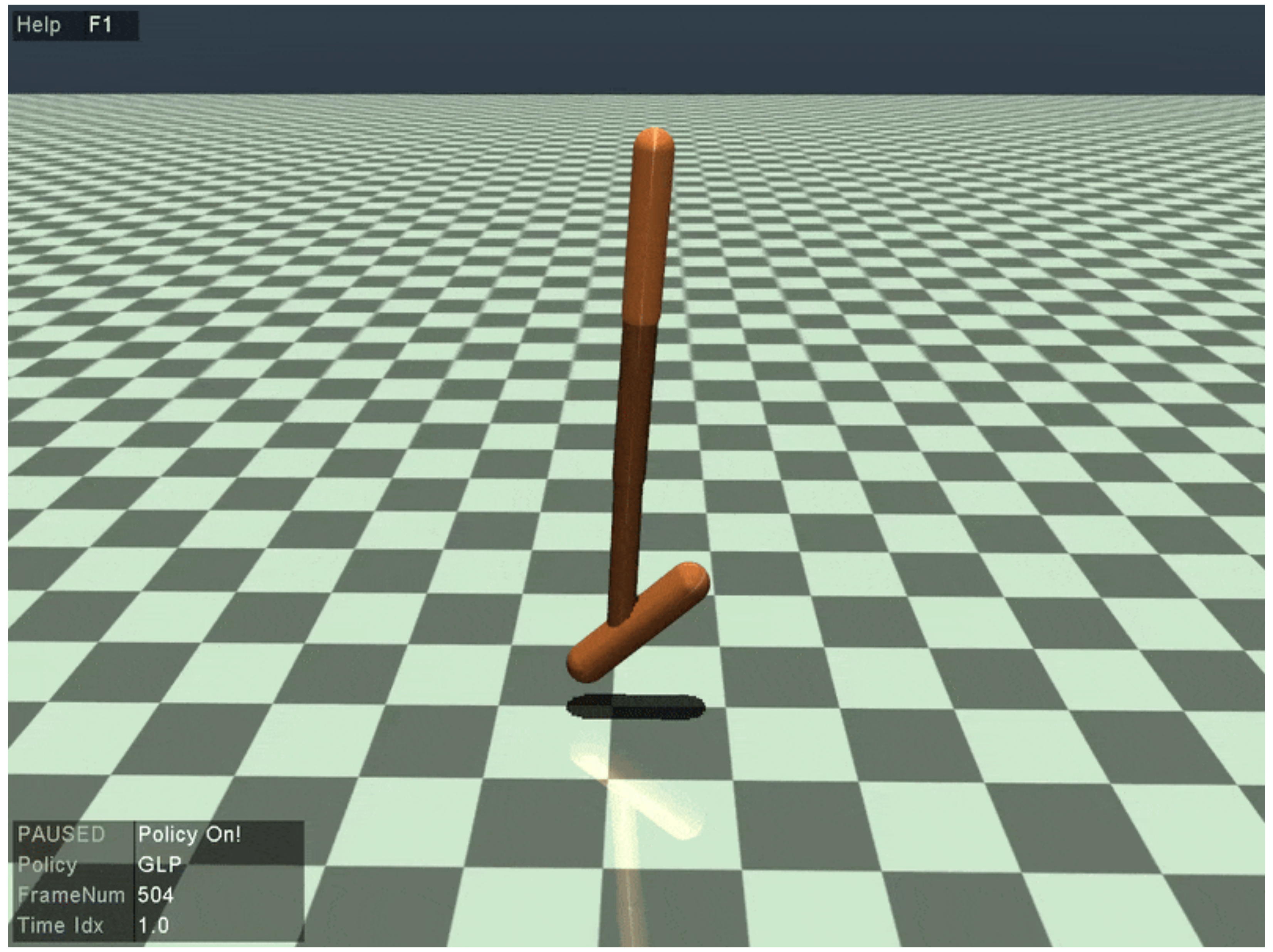
Let's examine the role of μ

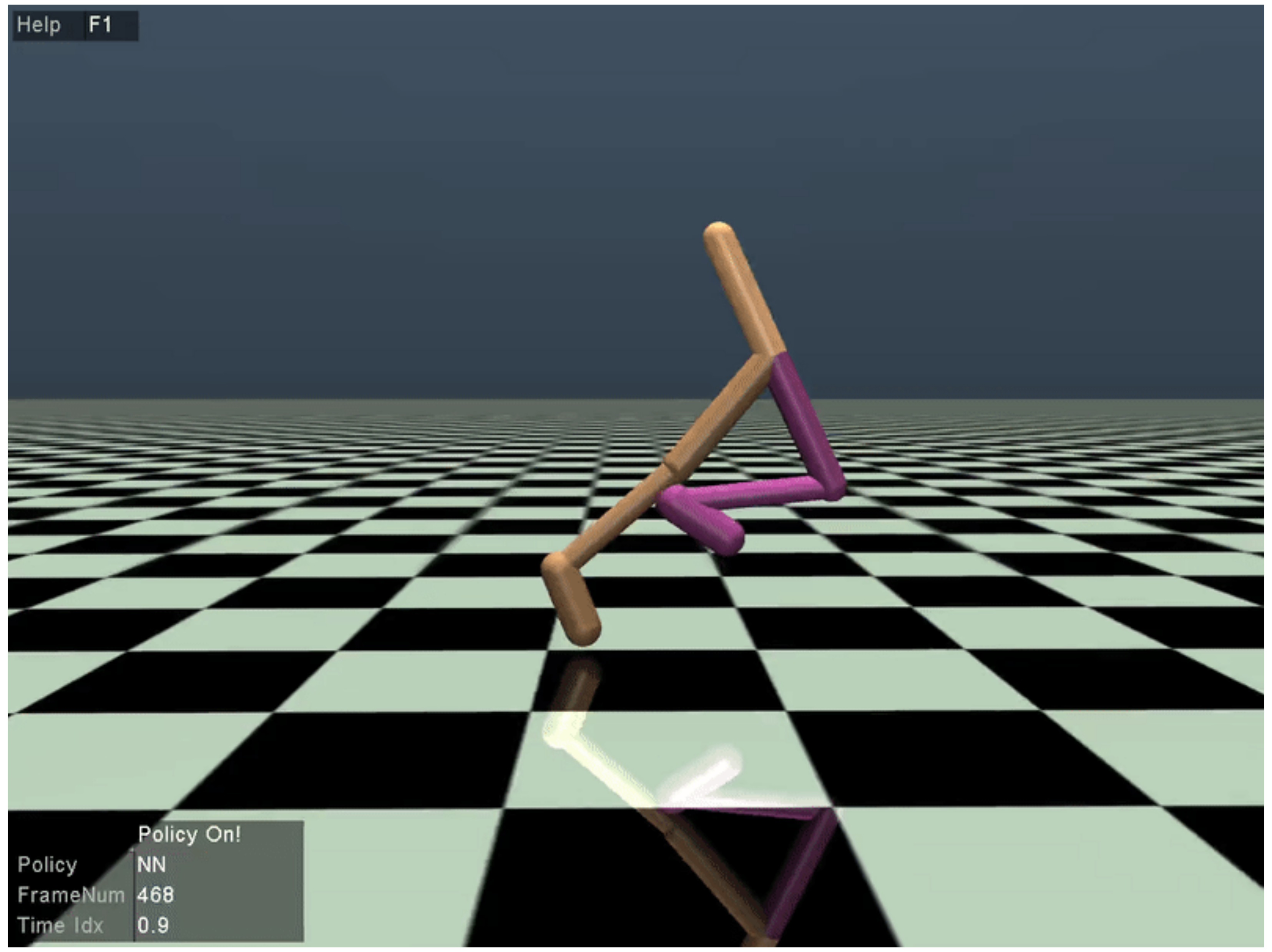


Thrun '92

- Suppose that somehow the distribution μ had better coverage.
  - e.g, if μ was uniform overall states in our toy problem, then all approaches we covered would work (with mild assumptions)
  - Theory: TRPO/NPG/PPO have better guarantees than fitted DP methods (assuming some "coverage")
- Strategies without coverage:
  - If we have a simulator, sometimes we can design μ to have better coverage.
    - this is helpful for robustness as well.
  - Imitation learning (next time).
    - An expert gives us samples from a "good" μ.
  - Explicit exploration:
    - UCB-VI: we'll merge two good ideas!
    - Encourage exploration in PG methods.
  - Try with reward shaping

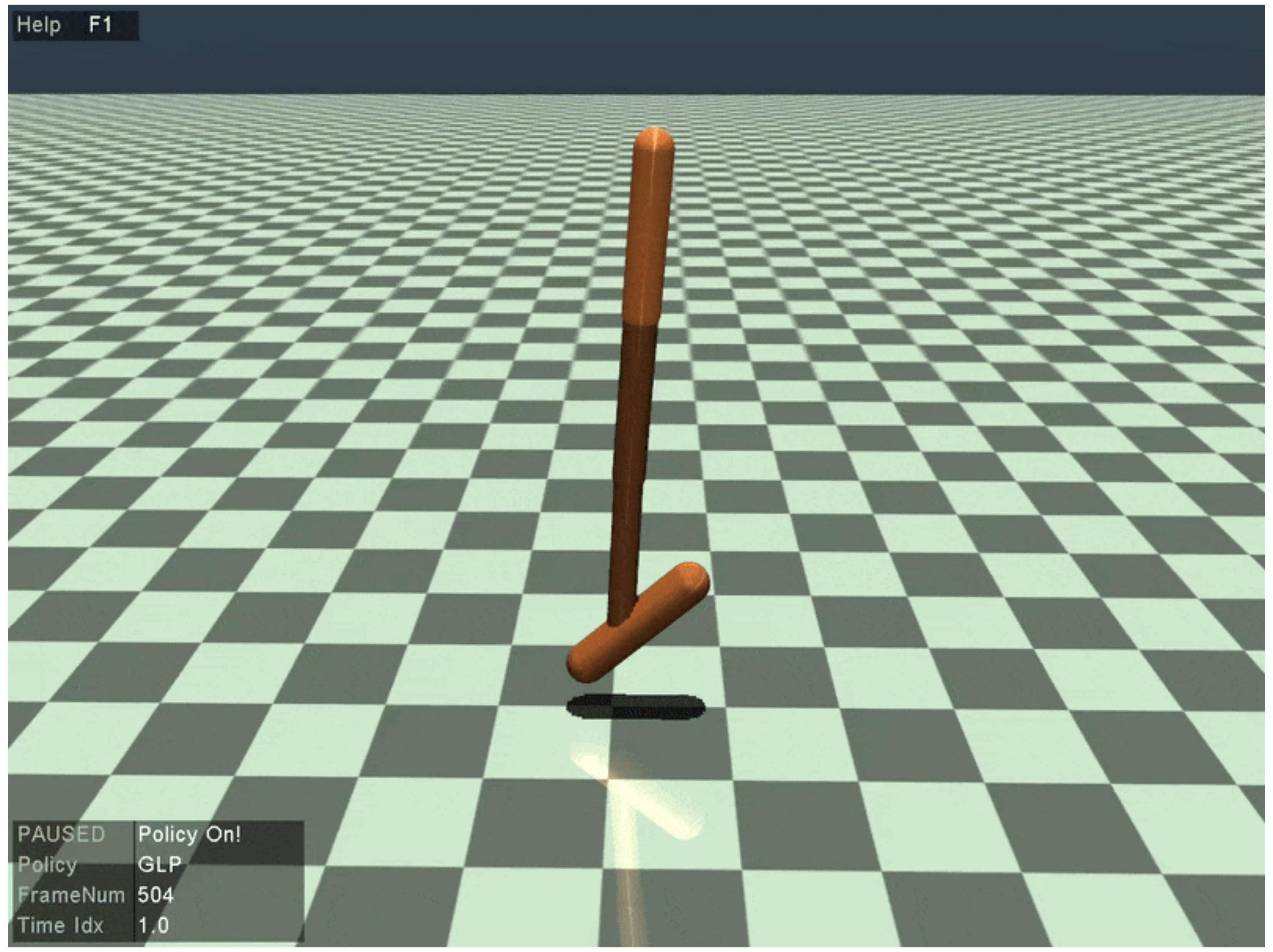
Aside: Brittle policies if we train starting from only from one configuration!

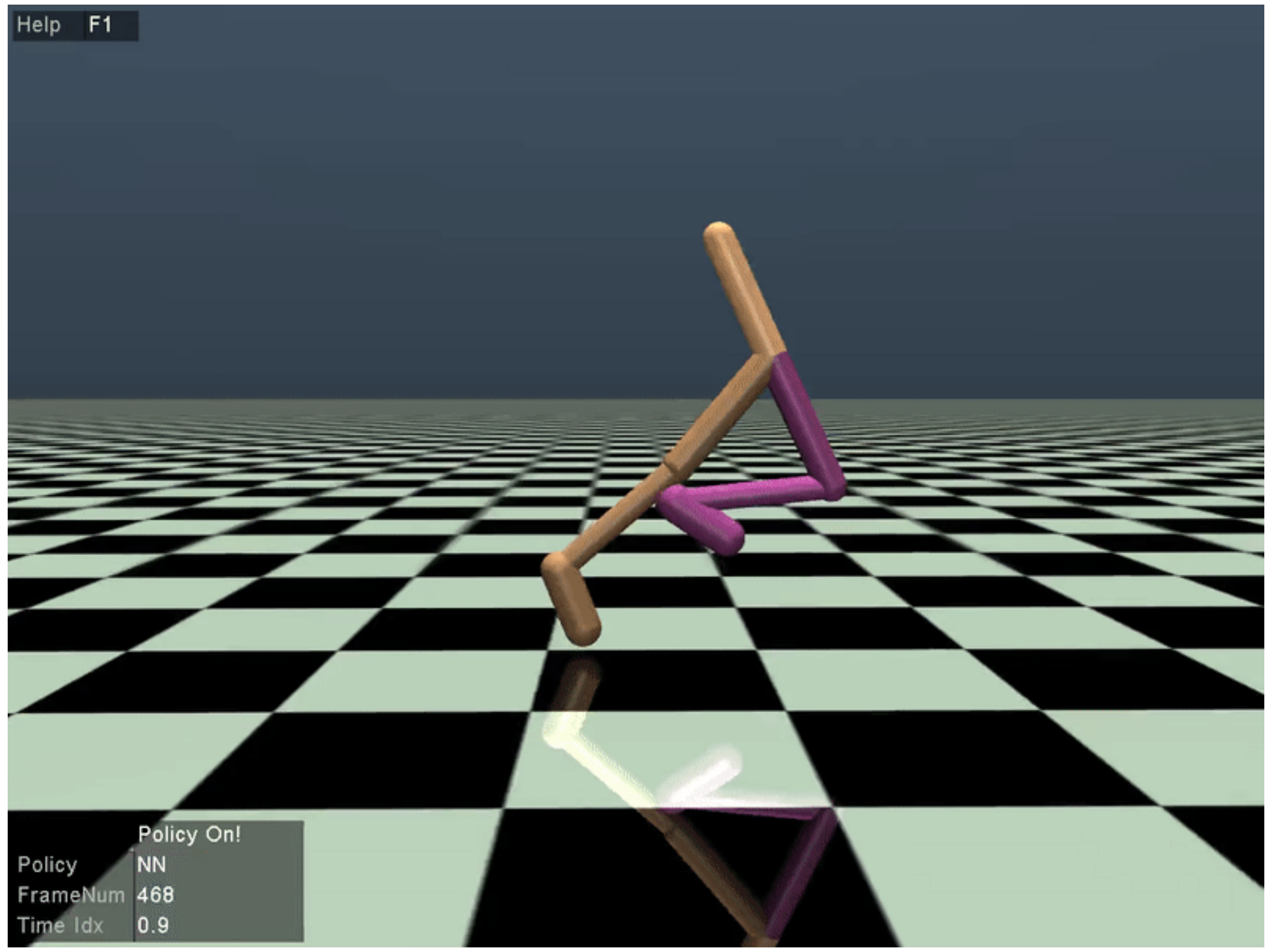




- [Rajeswaran, Lowrey, Todorov, K. 2017]: showed policies optimized for a single starting configuration s_0 are not robust!

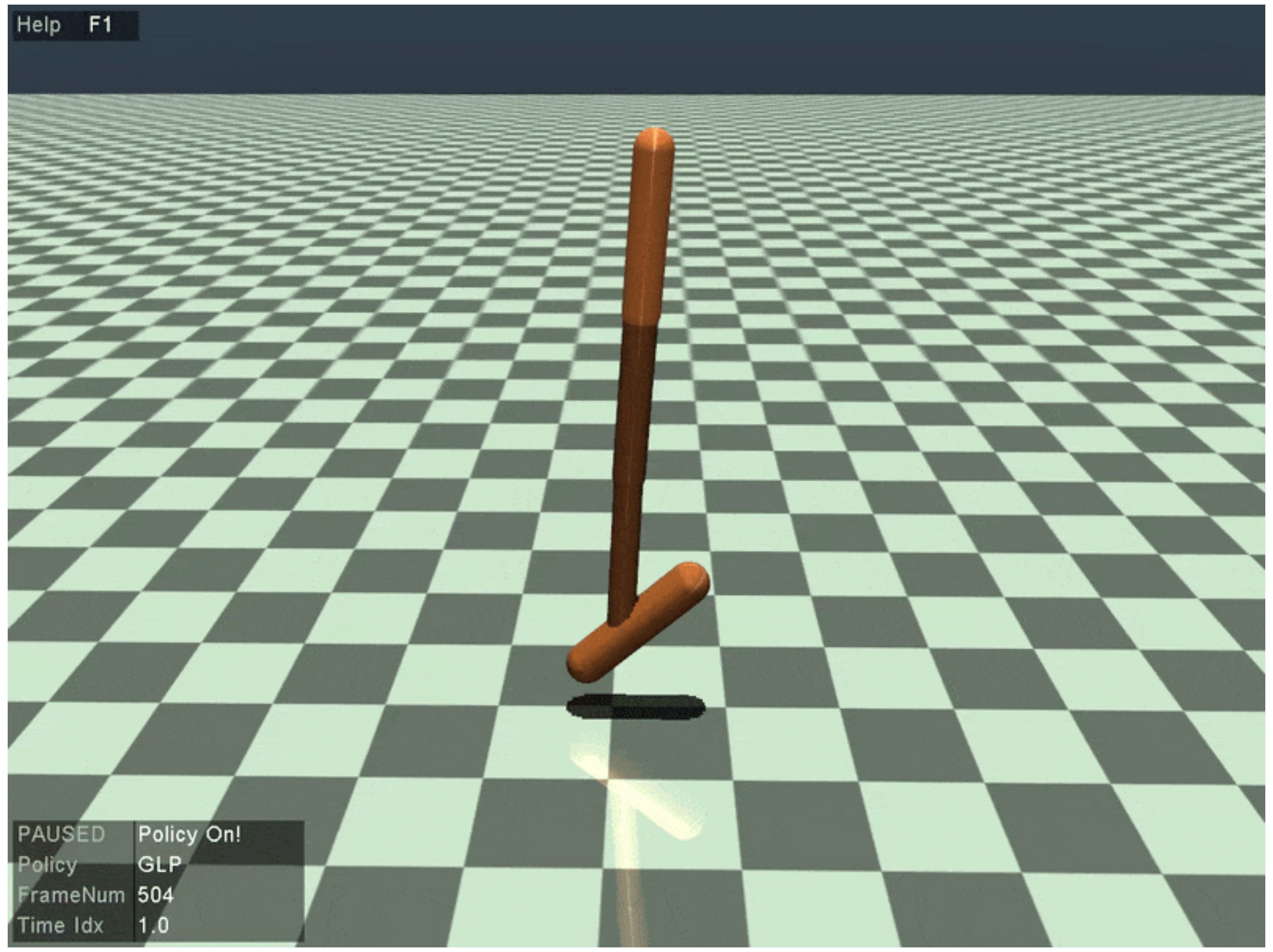
Aside: Brittle policies if we train starting from only from one configuration!

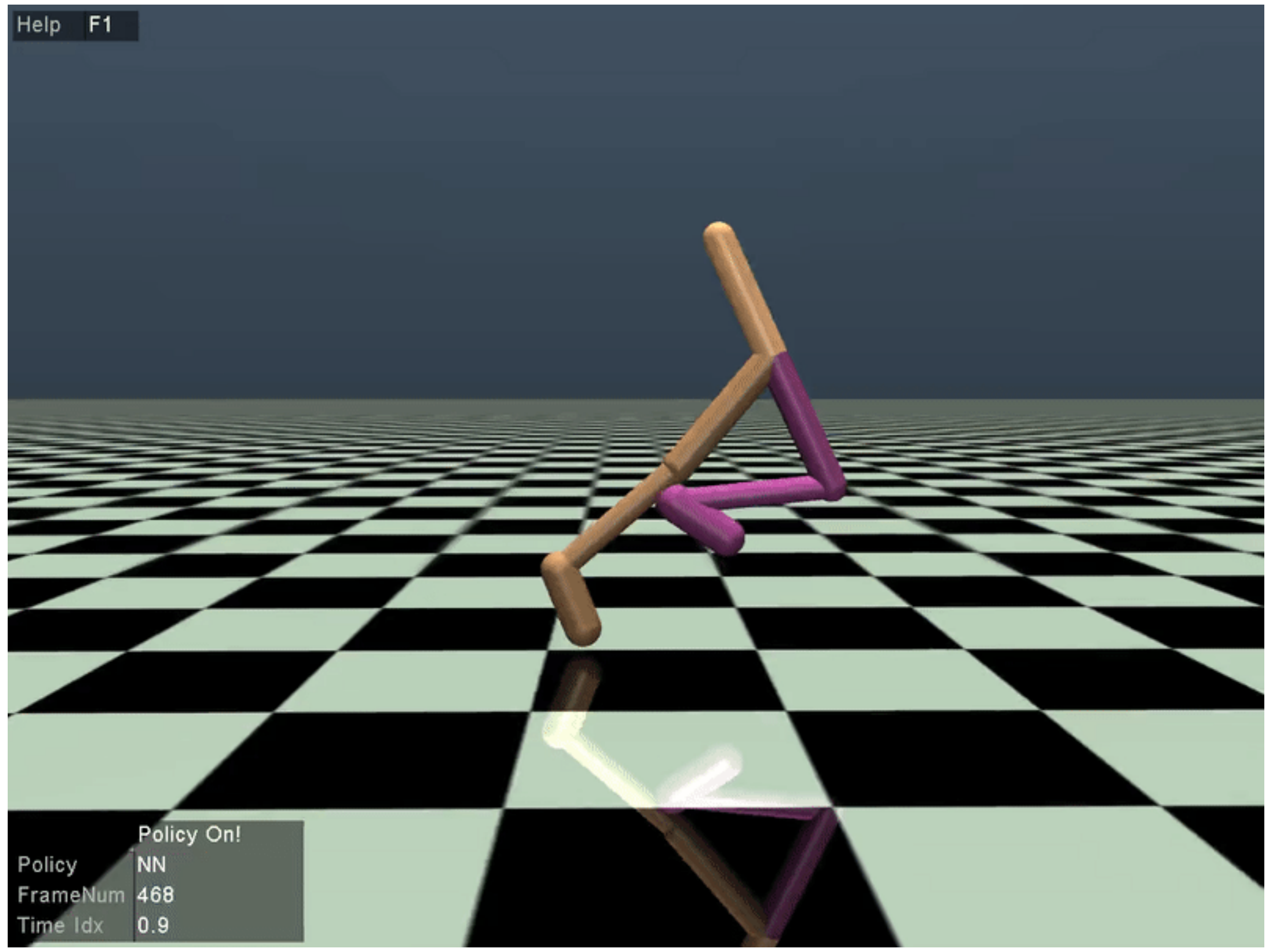




- [Rajeswaran, Lowrey, Todorov, K. 2017]: showed policies optimized for a single starting configuration s_0 are not robust!

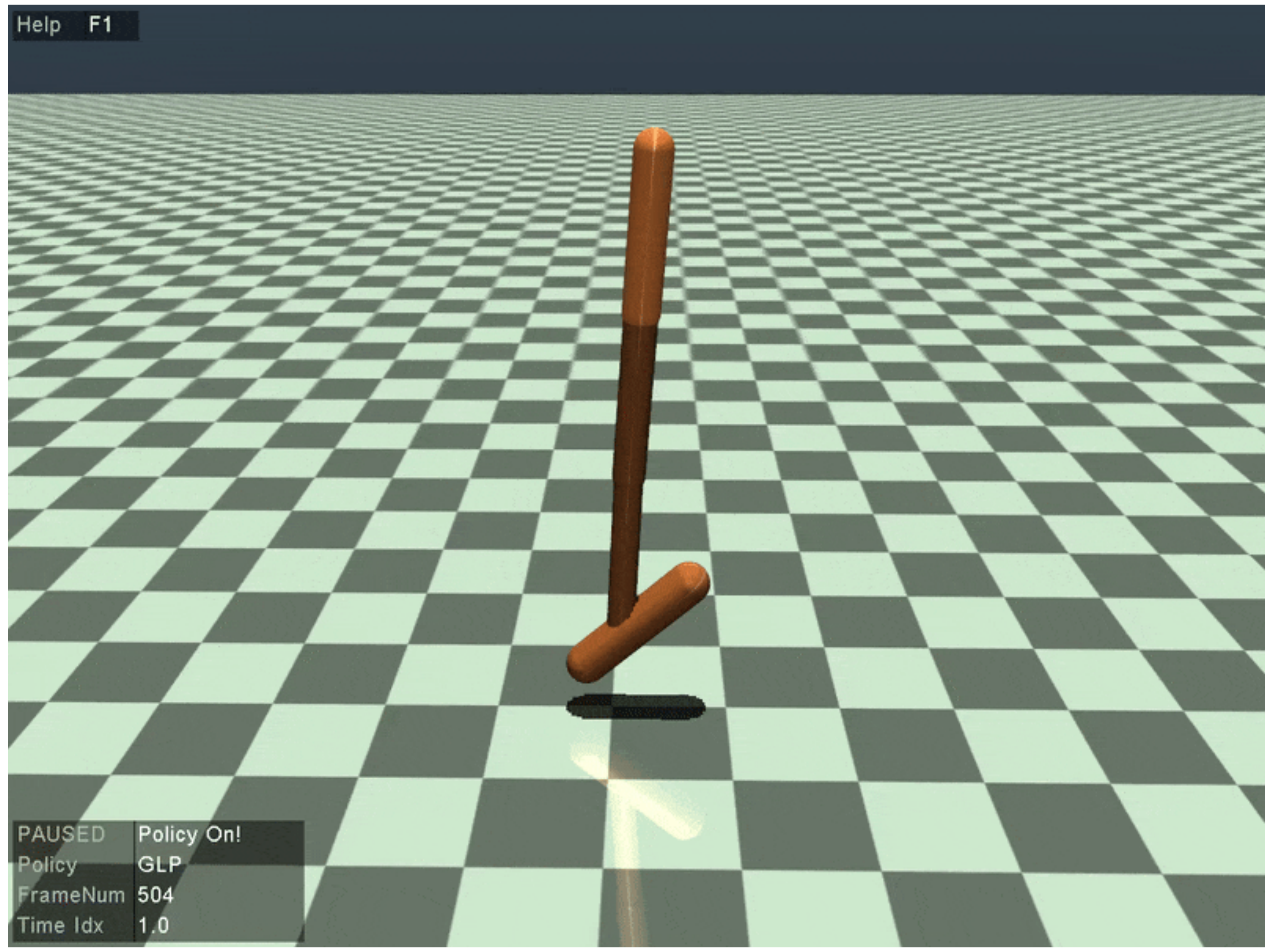
Aside: Brittle policies if we train starting from only from one configuration!

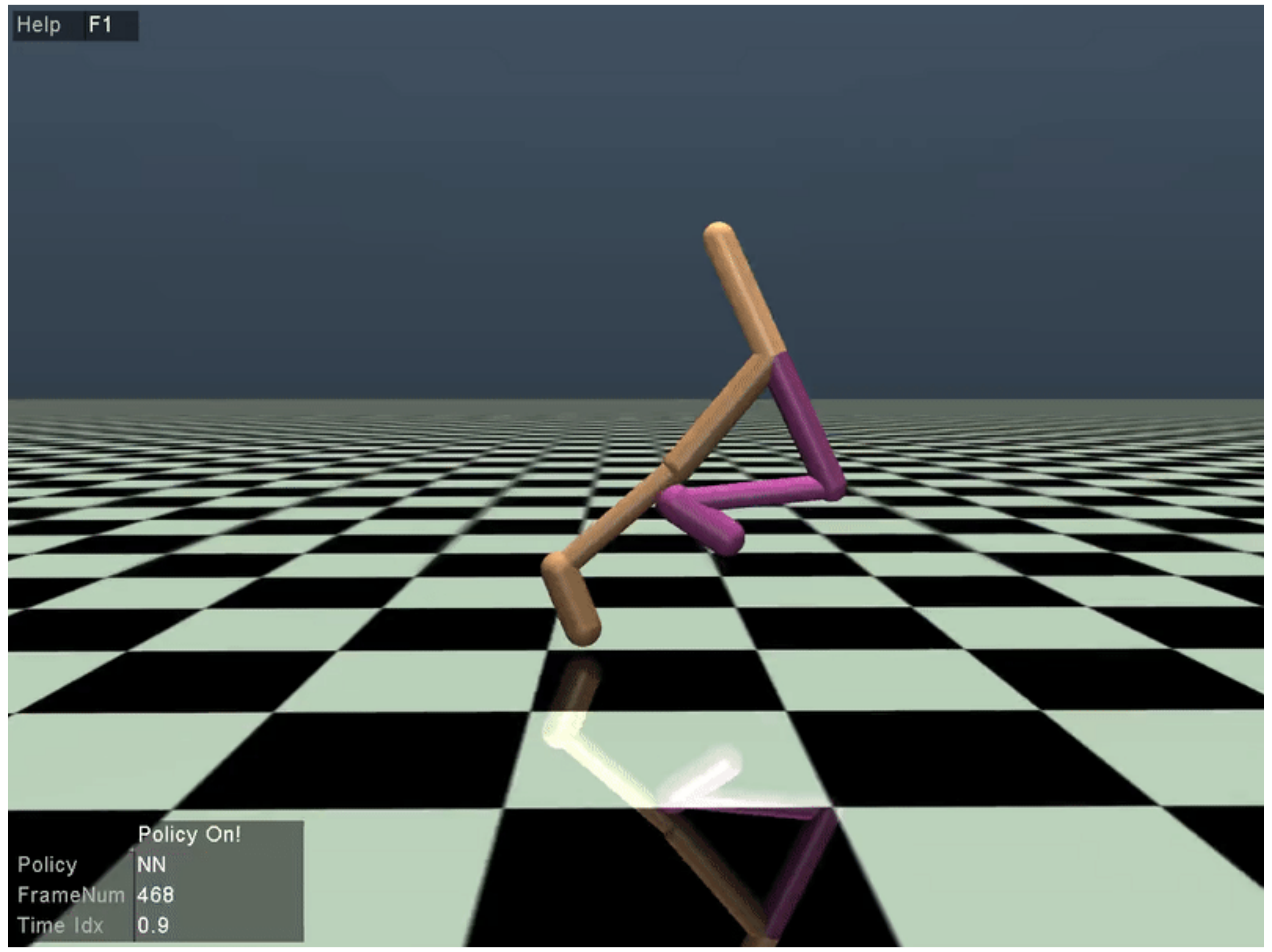




- [Rajeswaran, Lowrey, Todorov, K. 2017]: showed policies optimized for a single starting configuration s_0 are not robust!

Aside: Brittle policies if we train starting from only from one configuration!





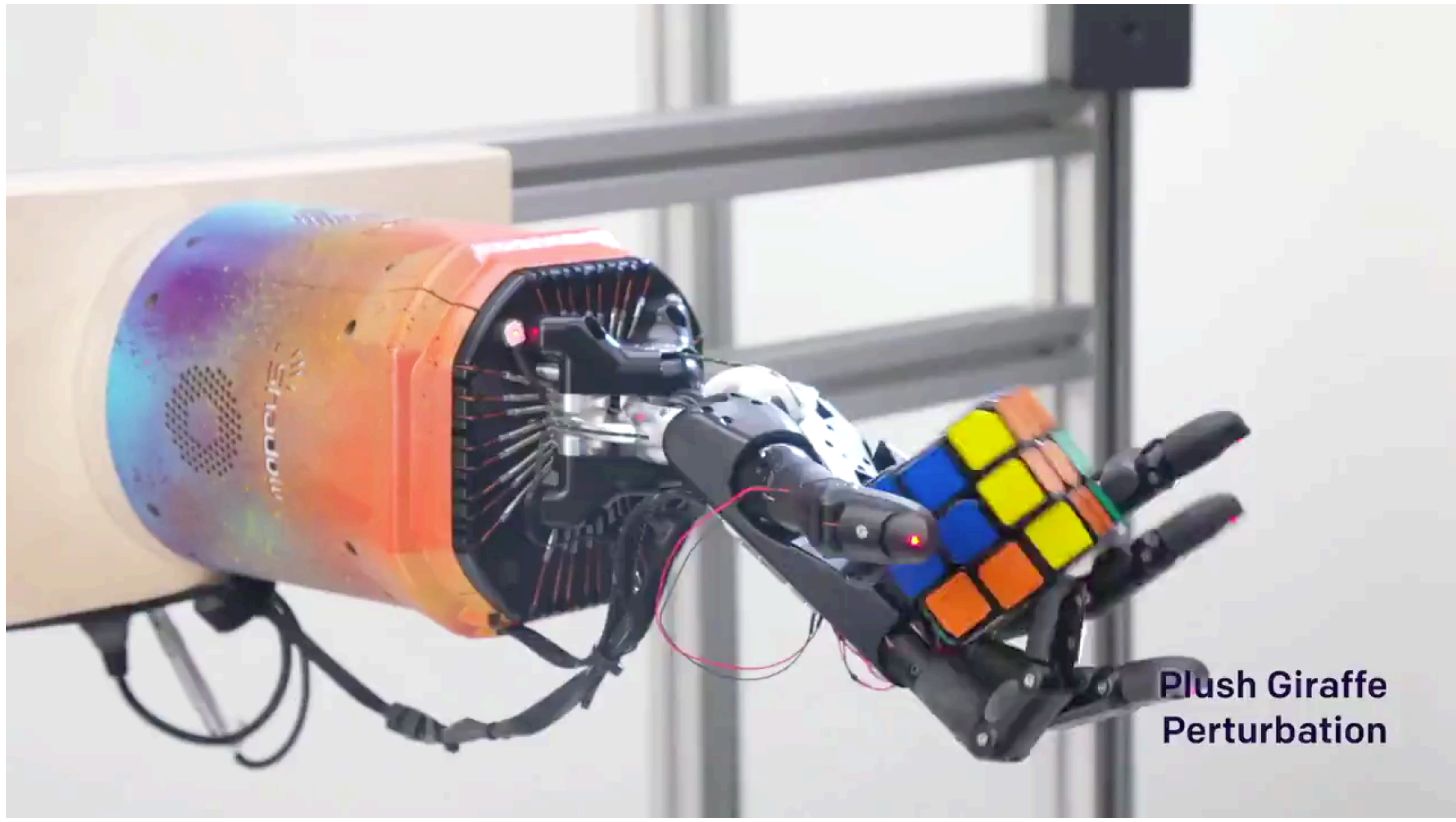
- [Rajeswaran, Lowrey, Todorov, K. 2017]: showed policies optimized for a single starting configuration s_0 are not robust!
- How to fix this?
  - Training from different starting configurations sampled from $s_0 \sim \mu$ fixes this:

$$\max_{\theta} \mathbb{E}_{s_0 \sim \mu}[V^{\theta}(s_0)]$$

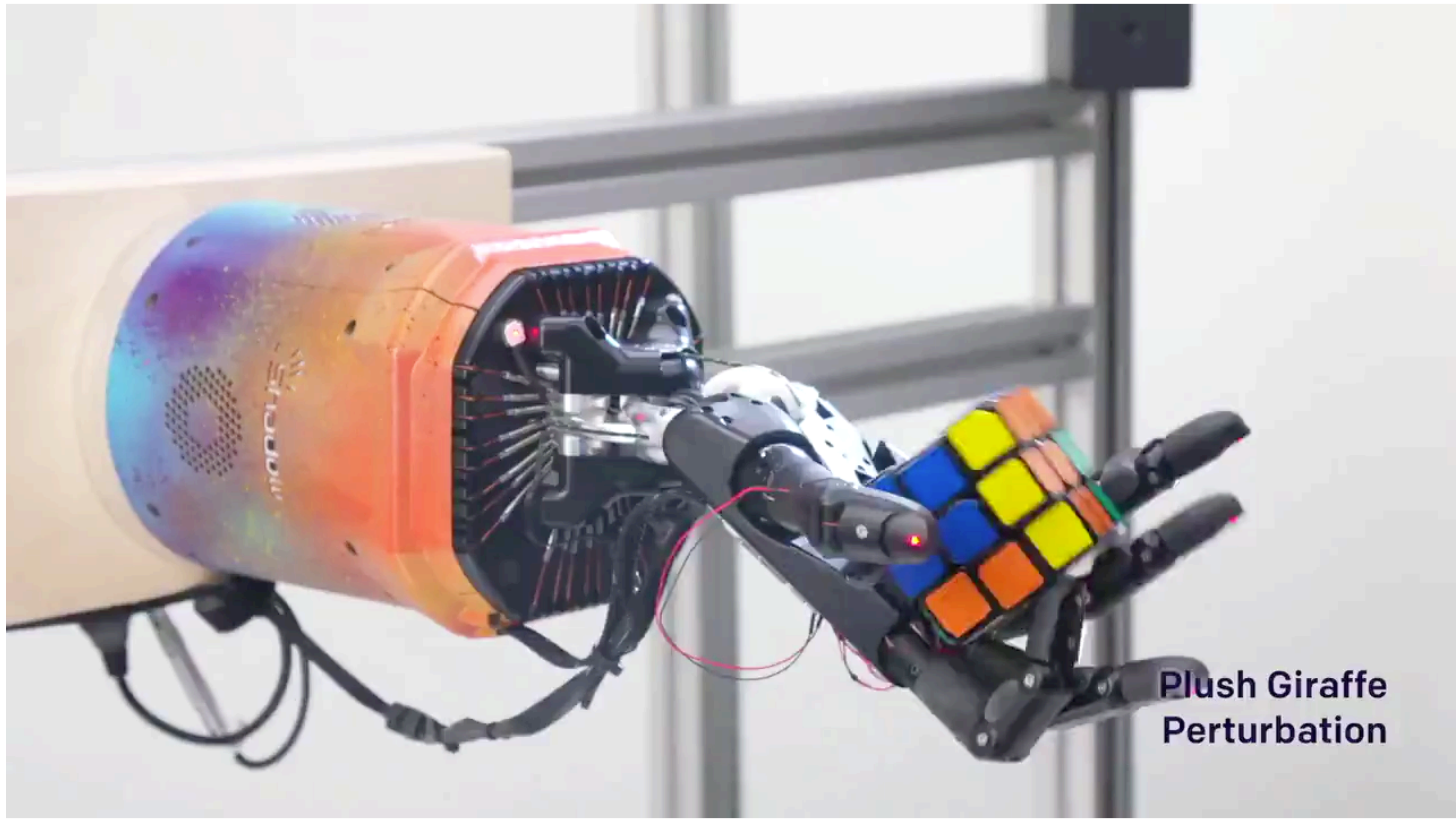
Even if starting position concentrated at just one point—good for robustness!

OpenAI: progress on dexterous hand manipulation

OpenAI: progress on dexterous hand manipulation



OpenAI: progress on dexterous hand manipulation



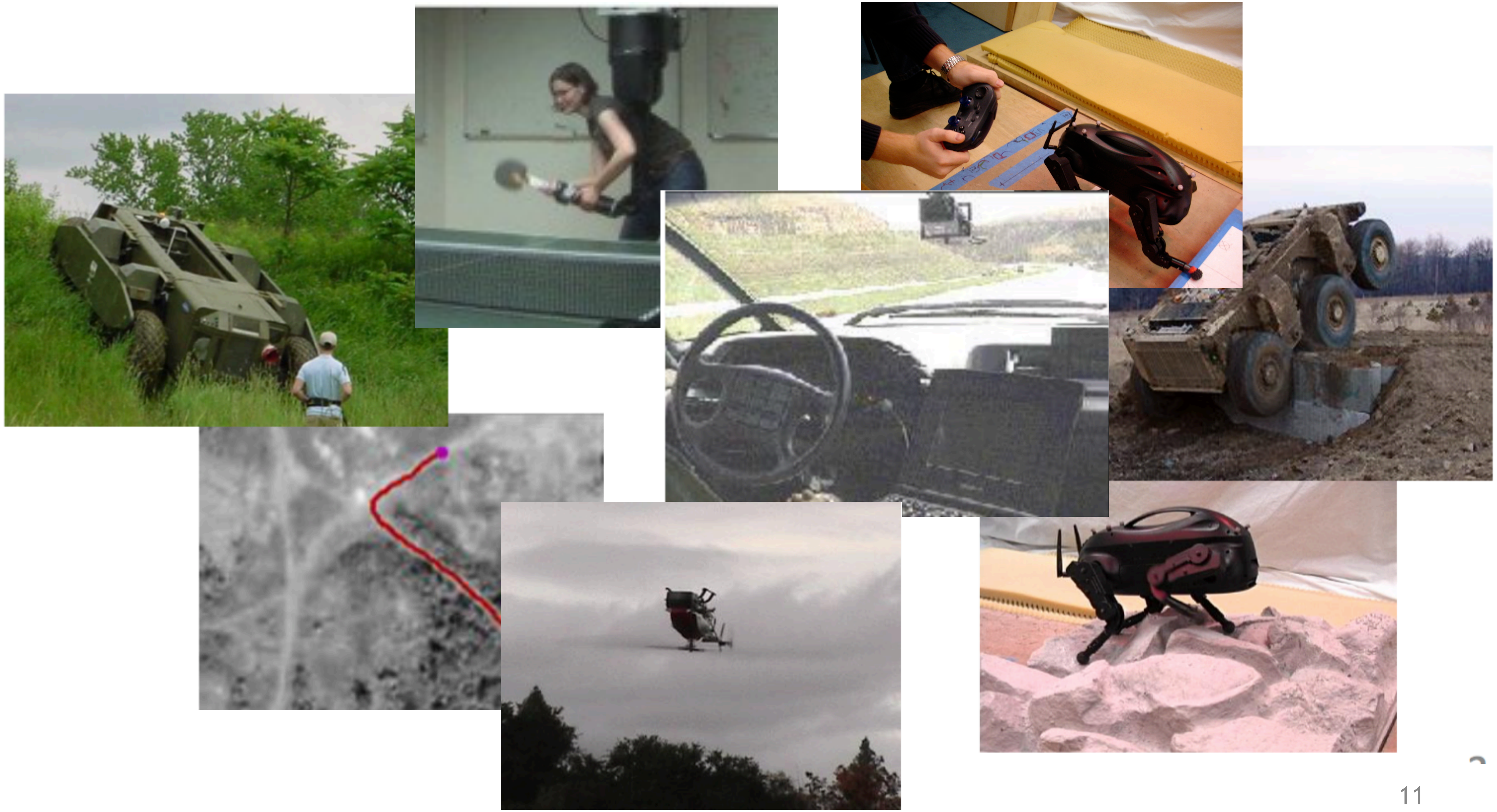
Trained with “domain randomization”

Basically, the measure $s_0 \sim \mu$ was diverse.

Today

- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

Imitation Learning



Imitation Learning

Imitation Learning

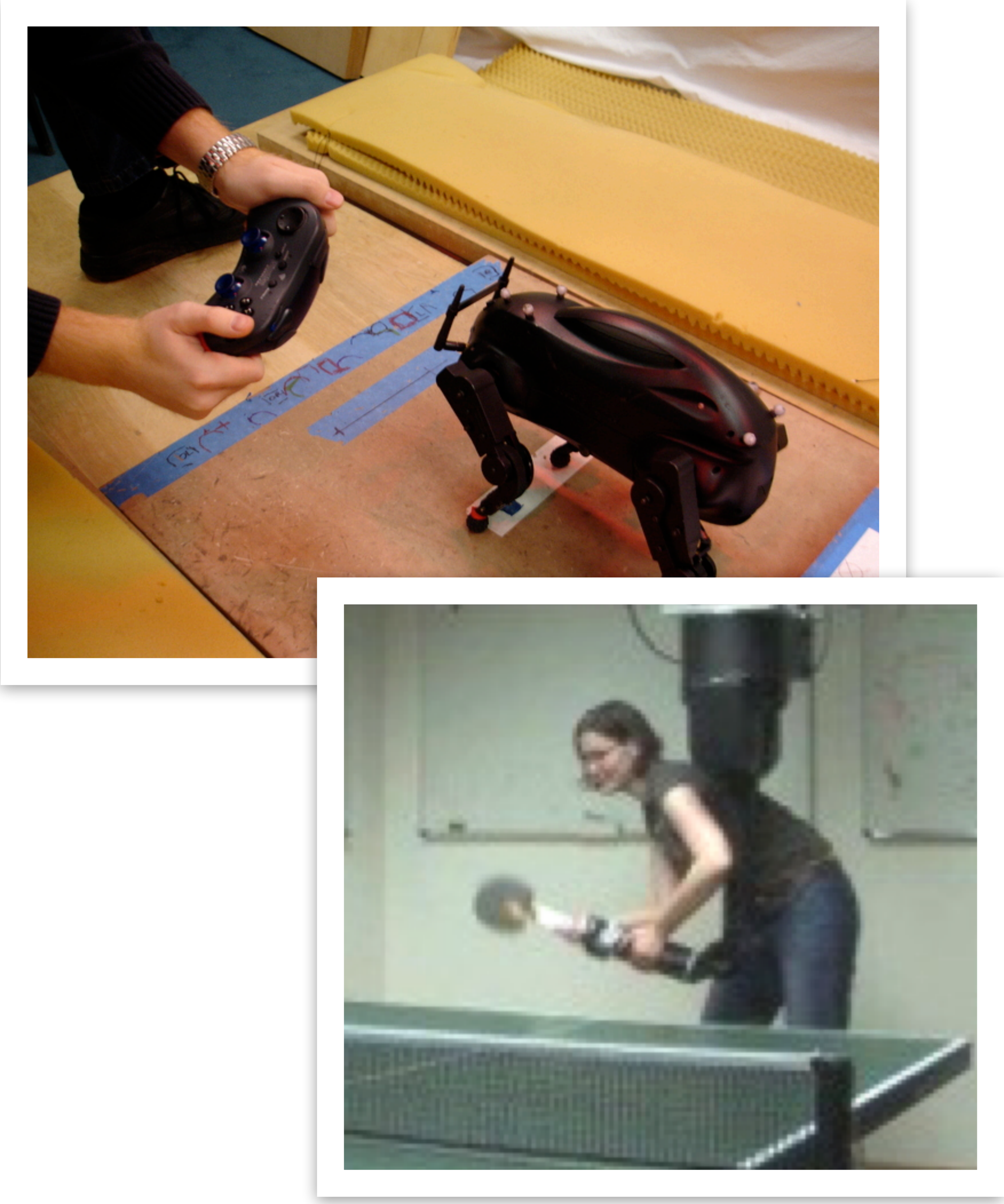
Imitation Learning

Expert Demonstrations

Imitation Learning

Expert Demonstrations

Machine Learning Algorithm



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

Imitation Learning

Expert Demonstrations

Machine Learning Algorithm

Policy π

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

Maps *states* to <u>actions</u>

Learning to Drive by Imitation

[Pomerleau89, Saxena05, Ross11a]

Input:

Camera Image

Policy

Output:

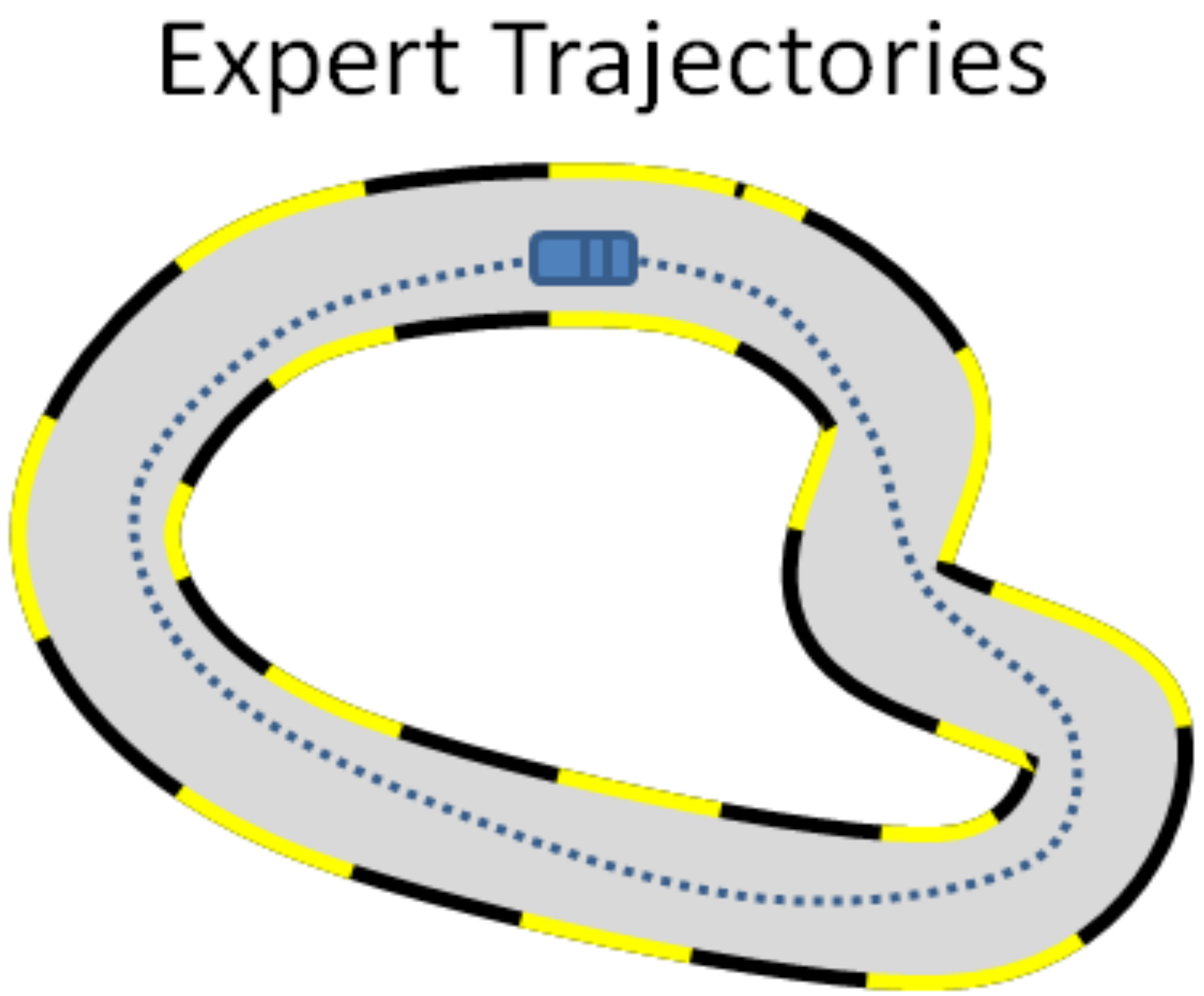
Steering Angle
in [-1, 1]

Today

- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

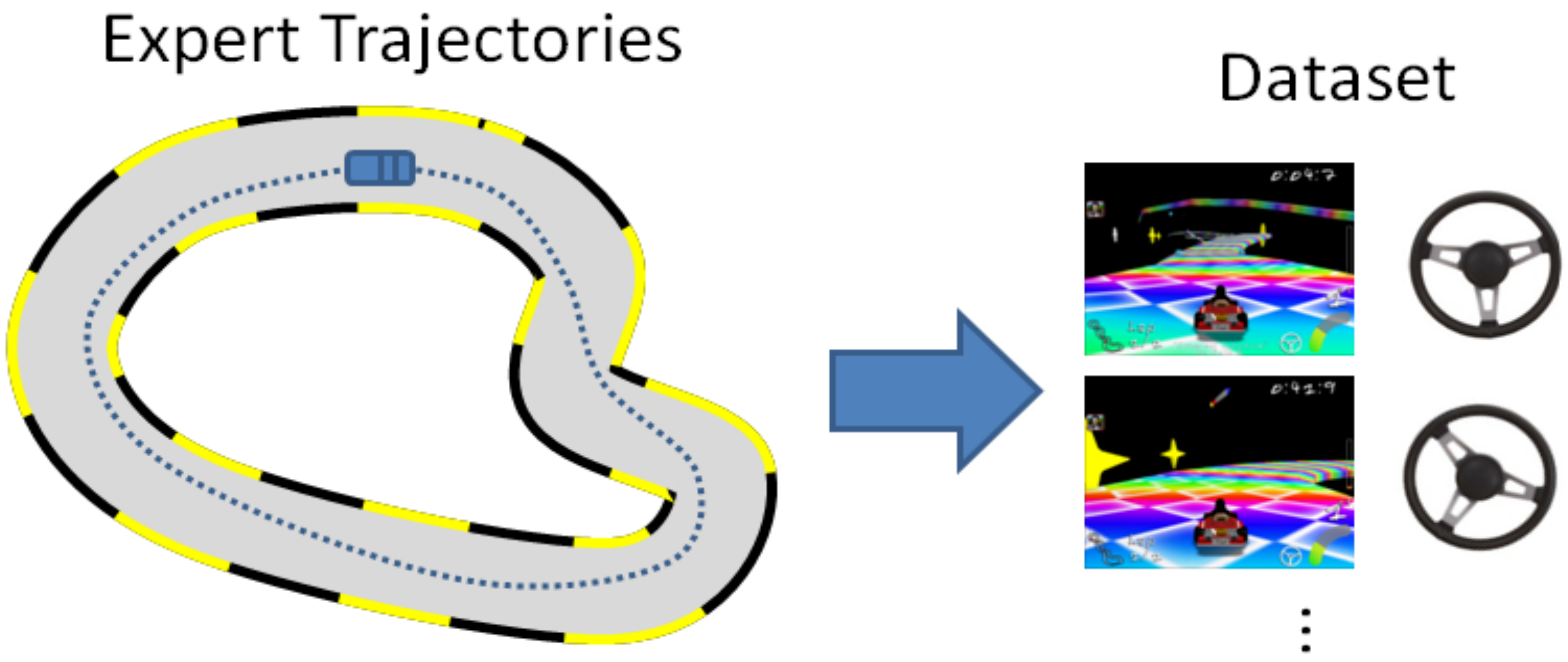
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



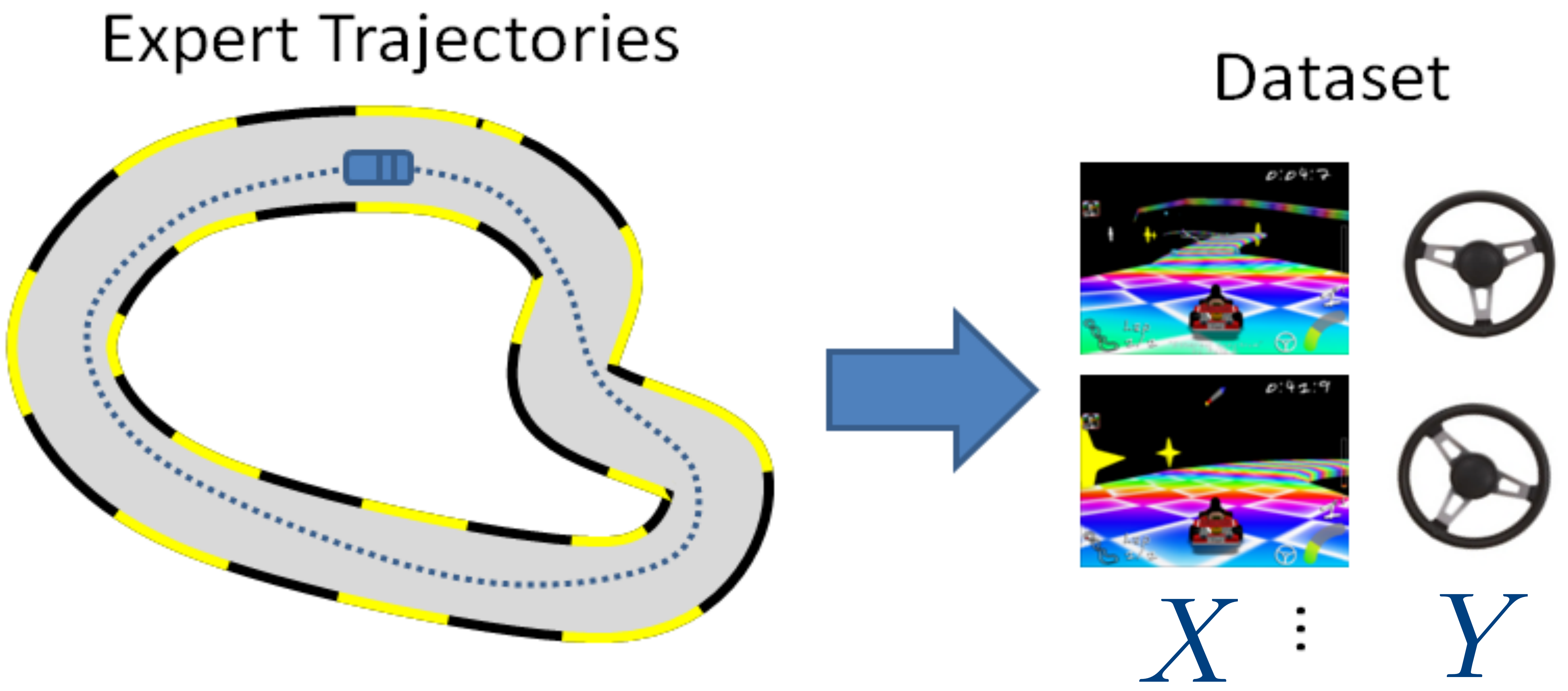
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



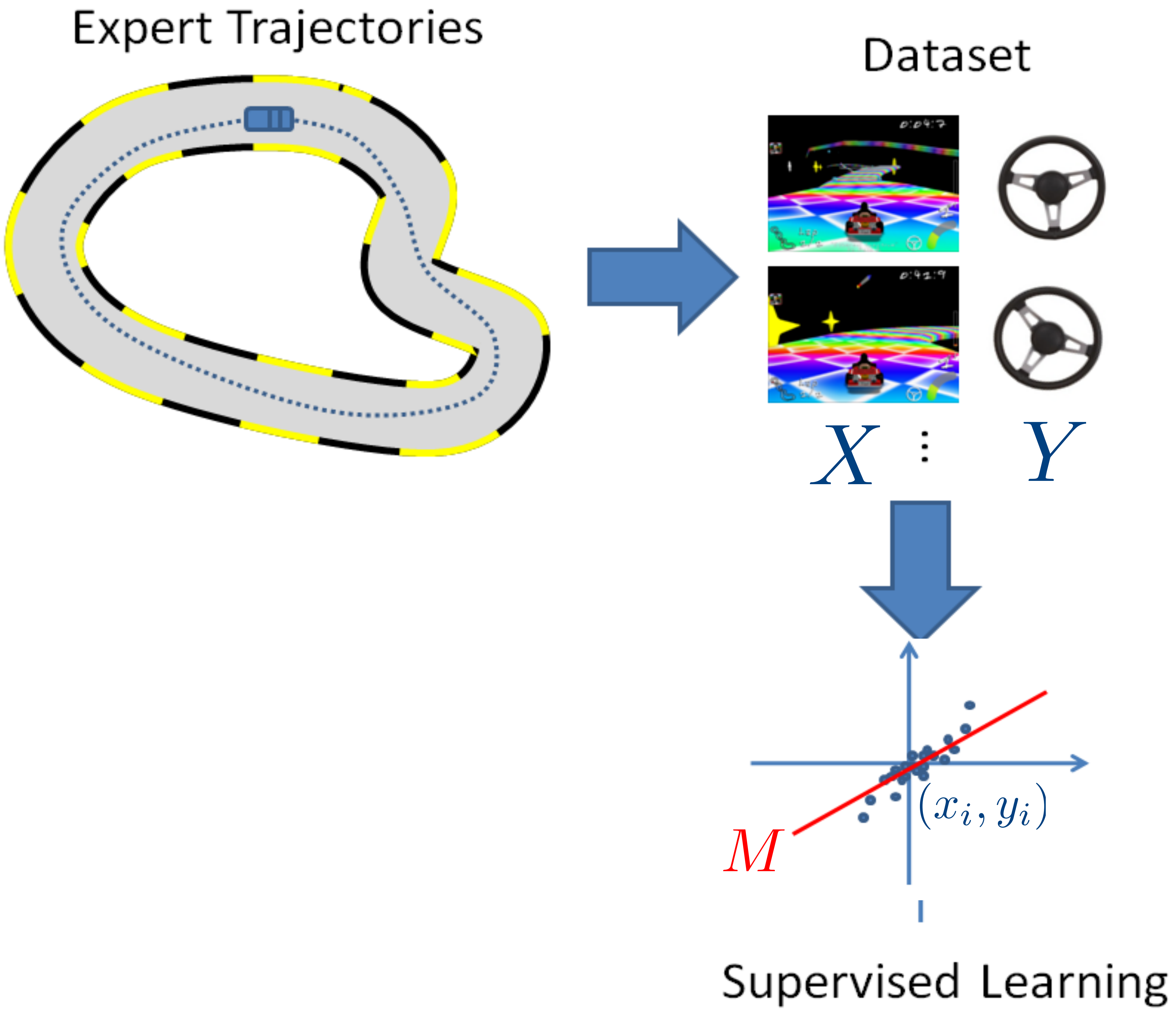
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



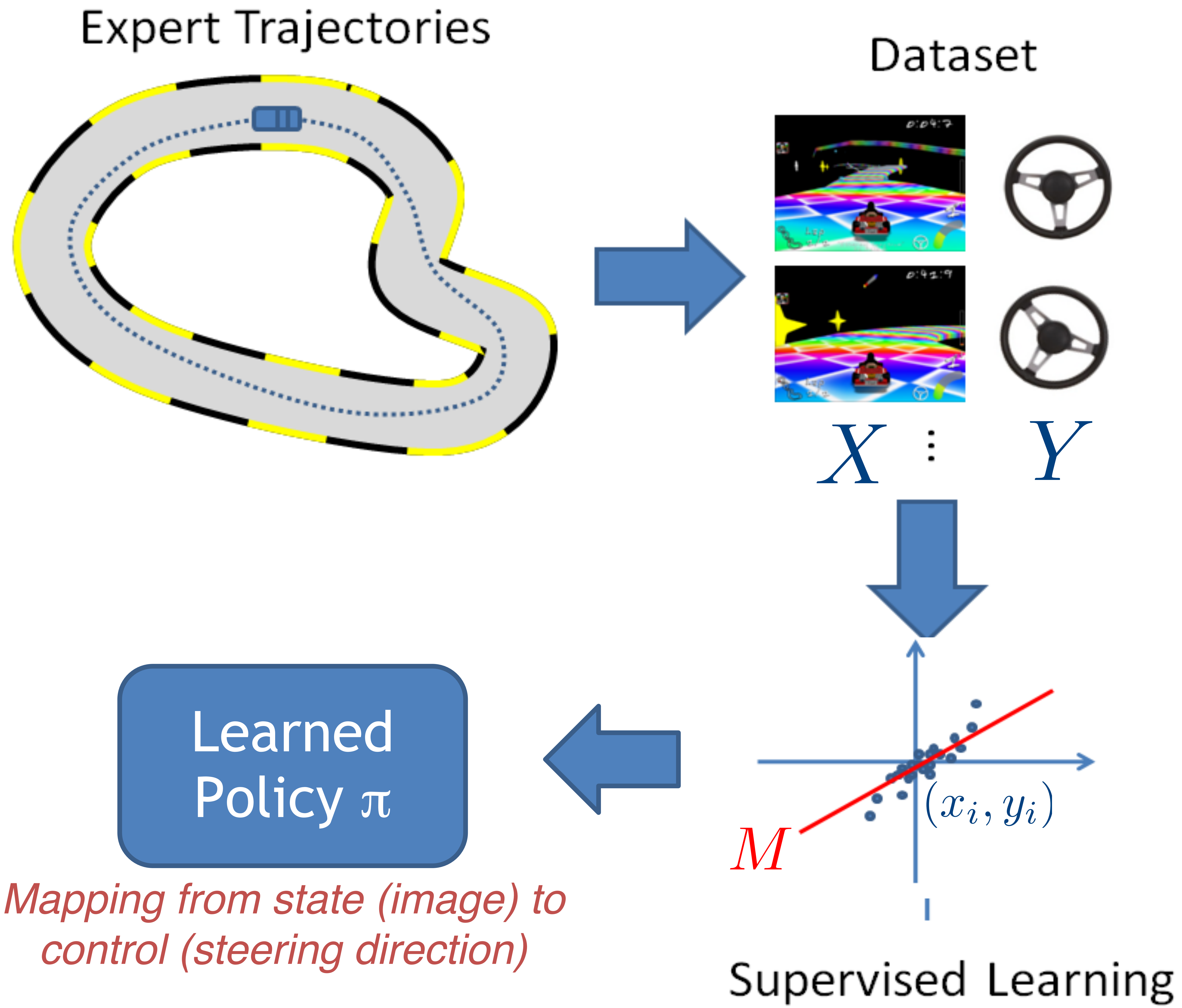
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning

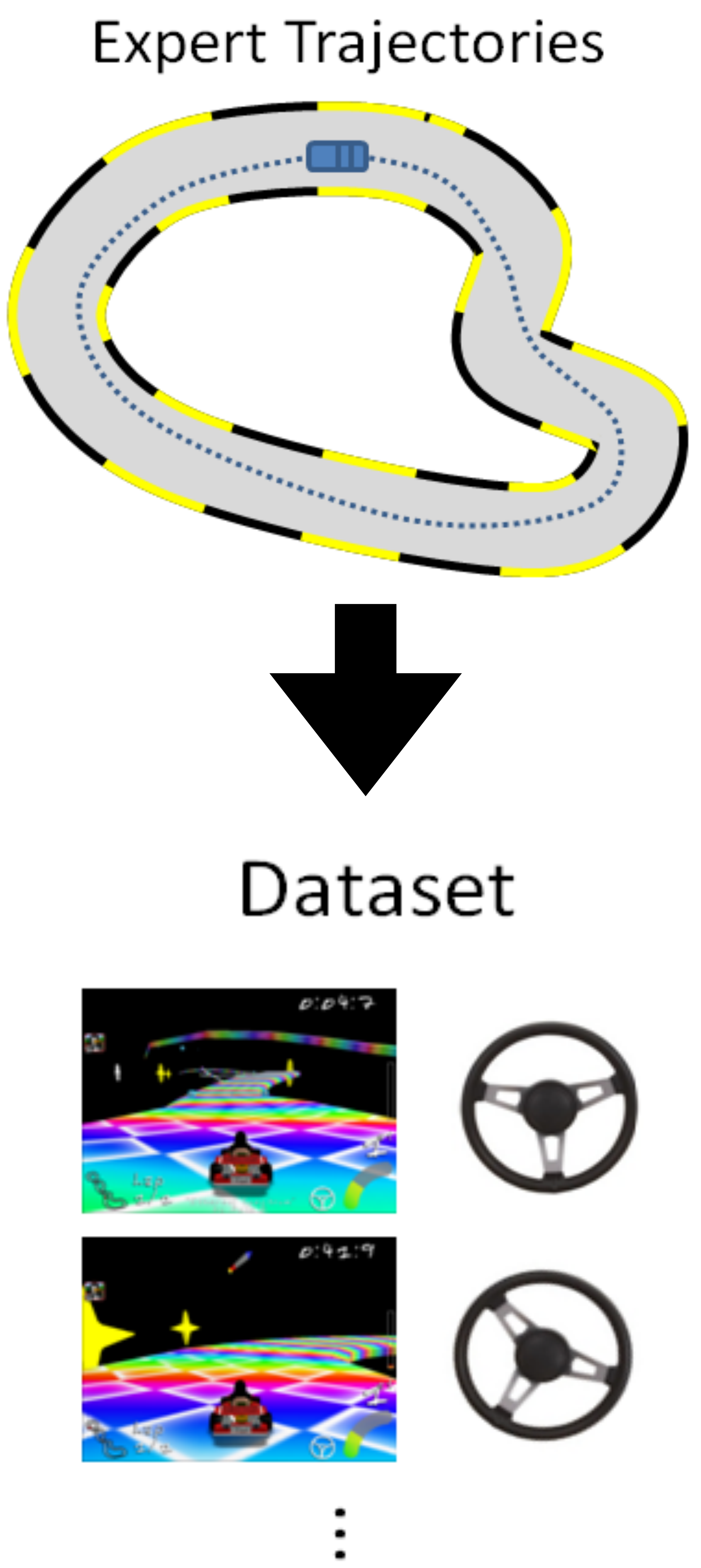


[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning

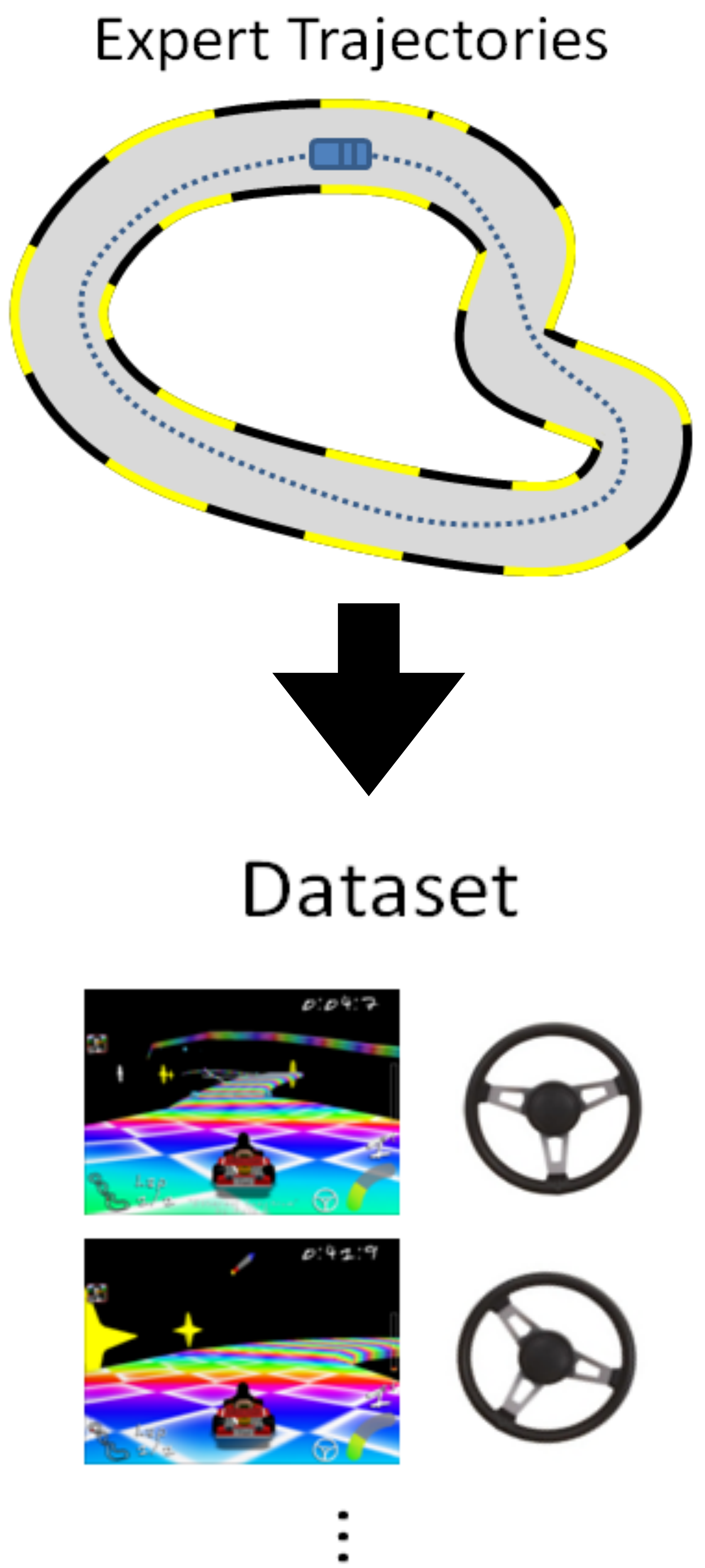


Let’s formalize the offline IL Setting and the Behavior Cloning algorithm



Finite horizon MDP $\mathscr{M}$

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

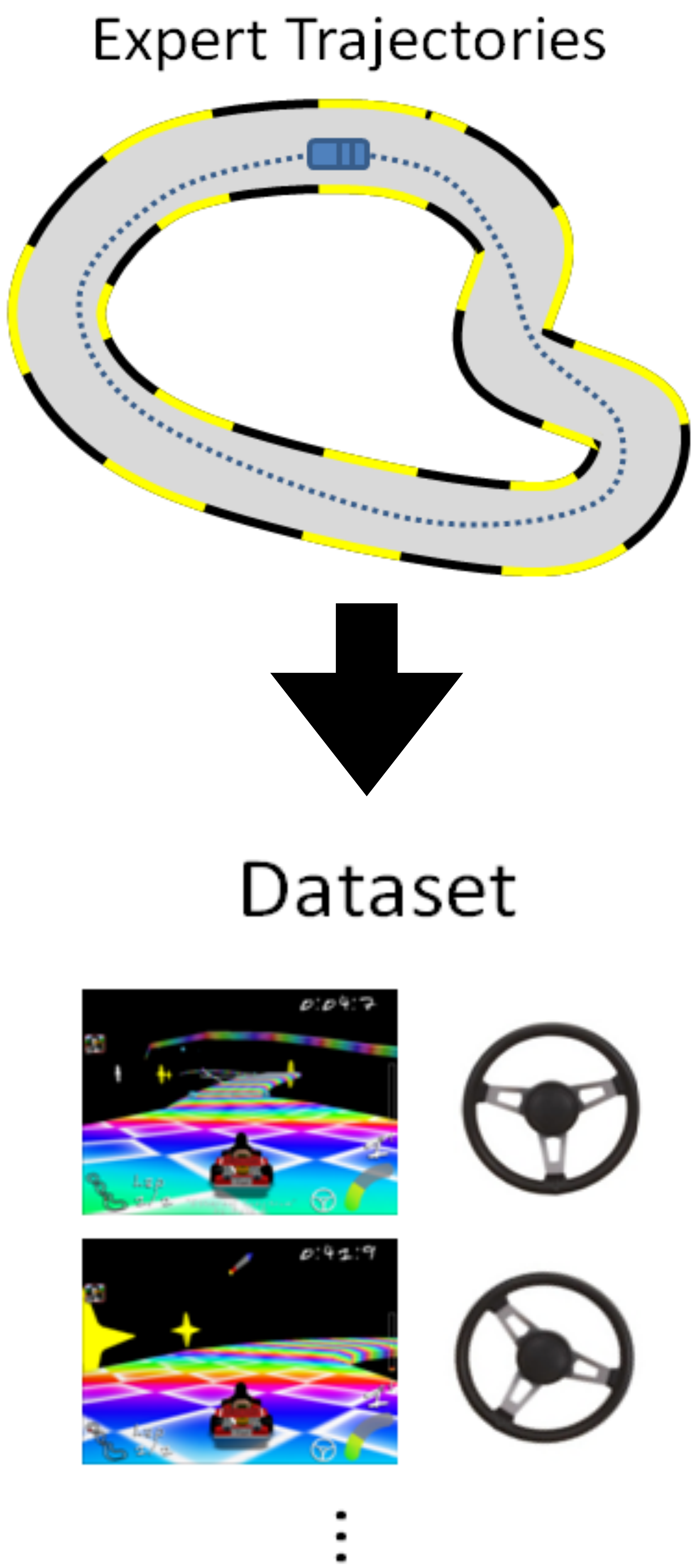


Finite horizon MDP $\mathscr{M}$

Ground truth reward $r(s, a) \in [0,1]$ is unknown;

Assume the expert has a good policy $\pi^\star$ (not necessarily opt)

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

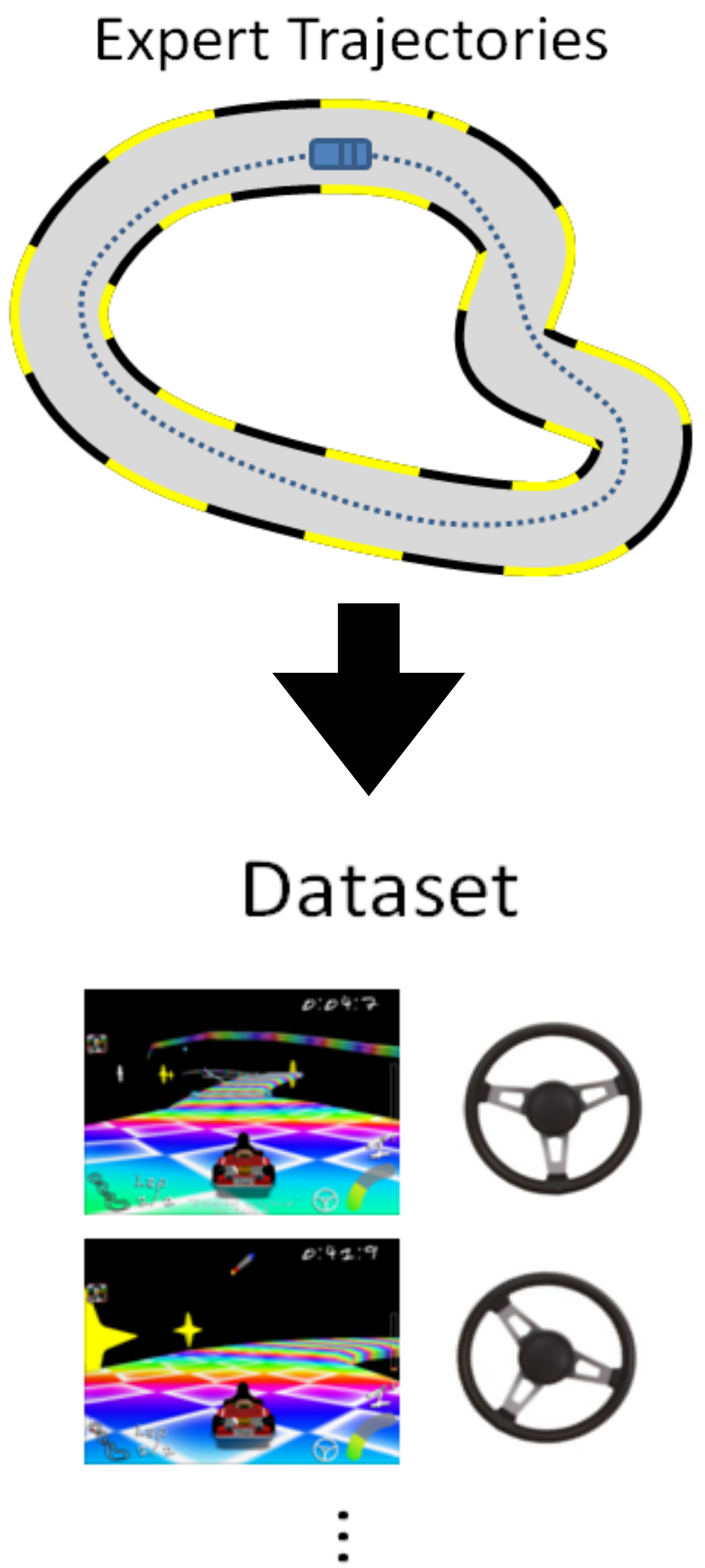


Finite horizon MDP $\mathscr{M}$

Ground truth reward $r(s, a) \in [0,1]$ is unknown;
Assume the expert has a good policy $\pi^\star$ (not necessarily opt)

We have a dataset of M trajectories: $\mathscr{D} = \{\tau_1, \ldots \tau_M\}$,
where $\tau_i = (s_h^i, a_h^i)_{h=0}^{H-1} \sim \rho_{\pi^\star}$

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



Finite horizon MDP $\mathscr{M}$

Ground truth reward $r(s, a) \in [0,1]$ is unknown;
Assume the expert has a good policy $\pi^\star$ (not necessarily opt)

We have a dataset of M trajectories: $\mathscr{D} = \{\tau_1, \ldots \tau_M\}$,
where $\tau_i = (s_h^i, a_h^i)_{h=0}^{H-1} \sim \rho_{\pi^\star}$

Goal: learn a policy from $\mathscr{D}$ that is as good as the expert $\pi^\star$

Let's formalize the Behavior Cloning (BC) algorithm

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

Let’s formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

Let's formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Let's formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

$\ell(\pi, s, a)$ is a loss function with many choices:

Let's formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

$\ell(\pi, s, a)$ is a loss function with many choices:

1. Classification (0/1) loss: $\mathbf{1}[\pi(s) \neq a]$

Let's formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

$\ell(\pi, s, a)$ is a loss function with many choices:

1. Classification (0/1) loss: $\mathbf{1}[\pi(s) \neq a]$
2. Negative log-likelihood (NLL): $\ell(\pi, s, a) = -\ln \pi(a \mid s)$

Let's formalize the Behavior Cloning (BC) algorithm

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

BC is a Reduction to Supervised Learning:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

$\ell(\pi, s, a)$ is a loss function with many choices:

1. Classification (0/1) loss: $\mathbf{1}[\pi(s) \neq a]$

2. Negative log-likelihood (NLL): $\ell(\pi, s, a) = -\ln \pi(a \mid s)$

3. square loss (i.e., regression for continuous action): $\ell(\pi, s, a) = \|\pi(s) - a\|_2^2$

Theorem: IL is (almost) as easy as SL

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem: IL is (almost) as easy as SL

$$\widehat{\pi} = \arg\min_{\pi\in\Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem [BC Performance]:

suppose we assume supervised learning succeeds, with ϵ classification error:

$$\mathbb{E}_{\tau\sim\rho_{\pi^\star}}\left[\frac{1}{H}\sum_{h=0}^{H-1} \mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h)\right]\right] \leq \epsilon,$$

Theorem: IL is (almost) as easy as SL

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem [BC Performance]:

suppose we assume supervised learning succeeds, with ϵ classification error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h)\right] \right] \leq \epsilon,$$

(where $\pi^\star$ is the expert policy, which need not be optimal)

Theorem: IL is (almost) as easy as SL

$$\widehat{\pi} = \arg\min_{\pi\in\Pi} \sum_{i=1}^{M}\sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem [BC Performance]:

suppose we assume supervised learning succeeds, with ϵ classification error:

$$\mathbb{E}_{\tau\sim\rho_{\pi^\star}}\left[\frac{1}{H}\sum_{h=0}^{H-1}\mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h)\right]\right] \leq \epsilon,$$

(where $\pi^\star$ is the expert policy, which need not be optimal)

then we have:

$$|V^{\pi^\star} - V^{\widehat{\pi}}| \leq ?$$

Theorem: IL is (almost) as easy as SL

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem [BC Performance]:

suppose we assume supervised learning succeeds, with ϵ classification error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1}\left[\hat{\pi}(s_h) \neq \pi^\star(s_h)\right] \right] \leq \epsilon,$$

(where $\pi^\star$ is the expert policy, which need not be optimal)

then we have:

$$|V^{\pi^\star} - V^{\hat{\pi}}| \leq ?$$

$$H^2 \epsilon$$

Theorem: IL is (almost) as easy as SL

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \sum_{h=0}^{H-1} \ell\left(\pi, s_h^i, a_h^i\right)$$

Note a training and testing "mismatch"

Theorem [BC Performance]:

suppose we assume supervised learning succeeds, with ϵ classification error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h)\right] \right] \leq \epsilon,$$

(where $\pi^\star$ is the expert policy, which need not be optimal)

then we have:

$$| V^{\pi^\star} - V^{\widehat{\pi}} | \leq ?$$

$$H^2 \epsilon$$

The quadratic amplification is annoying

Proof:

Proof:

By the PDL

$$|V^{\pi^\star}(s) - V^{\widehat{\pi}}(s)| = \left|\mathbb{E}_{\tau \sim \rho_{\pi^\star}}\left[\sum_{h=0}^{H-1} A_h^{\widehat{\pi}}(s_h, a_h)\right]\right|$$

Proof:

By the PDL

$$|V^{\pi^\star}(s) - V^{\hat{\pi}}(s)| = \left| \mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\sum_{h=0}^{H-1} A_h^{\hat{\pi}}(s_h, a_h) \right] \right|$$

$$= \left| \mathbb{E}_{s_1, \ldots s_h \sim \rho_{\pi^\star}} \left[\sum_{h=0}^{H-1} A_h^{\hat{\pi}}(s_h, \pi^\star(s_h)) \right] \right|$$

Proof:

By the PDL

$$|V^{\pi^\star}(s) - V^{\widehat{\pi}}(s)| = \left| \mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\sum_{h=0}^{H-1} A_h^{\widehat{\pi}}(s_h, a_h) \right] \right|$$

$$= \left| \mathbb{E}_{s_1, \ldots s_h \sim \rho_{\pi^\star}} \left[\sum_{h=0}^{H-1} A_h^{\widehat{\pi}}(s_h, \pi^\star(s_h)) \right] \right|$$

$$\leq H \left| \mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\sum_{h=0}^{H-1} \mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h) \right] \right] \right|$$

Proof:

By the PDL

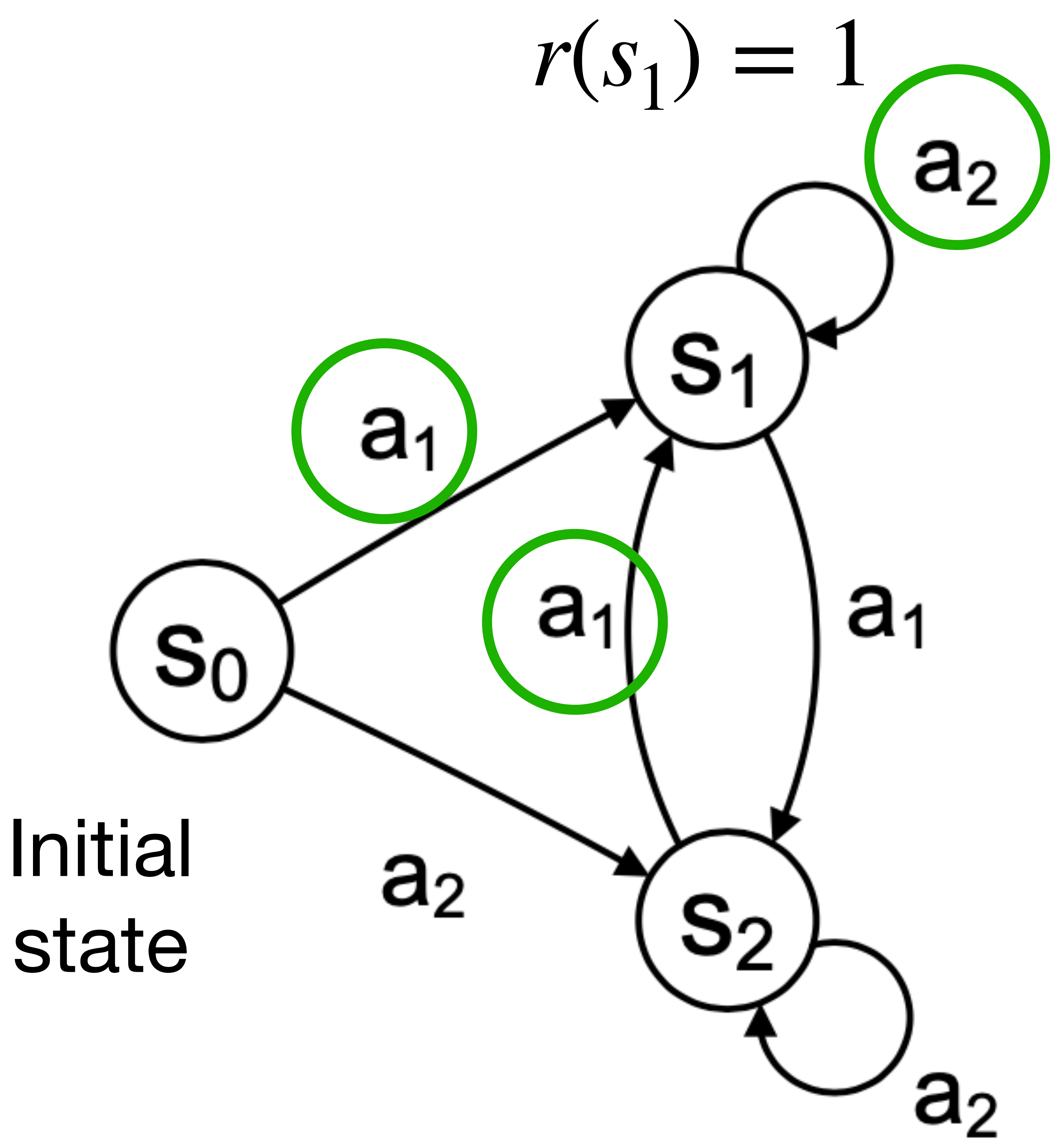
$$|V^{\pi^\star}(s) - V^{\hat{\pi}}(s)| = \left|\mathbb{E}_{\tau\sim\rho_{\pi^\star}}\left[\sum_{h=0}^{H-1} A_h^{\hat{\pi}}(s_h, a_h)\right]\right|$$

$$= \left|\mathbb{E}_{s_1,\ldots s_h\sim\rho_{\pi^\star}}\left[\sum_{h=0}^{H-1} A_h^{\hat{\pi}}(s_h, \pi^\star(s_h))\right]\right|$$

$$\leq H\left|\mathbb{E}_{\tau\sim\rho_{\pi^\star}}\left[\sum_{h=0}^{H-1} \mathbf{1}\left[\hat{\pi}(s_h) \neq \pi^\star(s_h)\right]\right]\right|$$

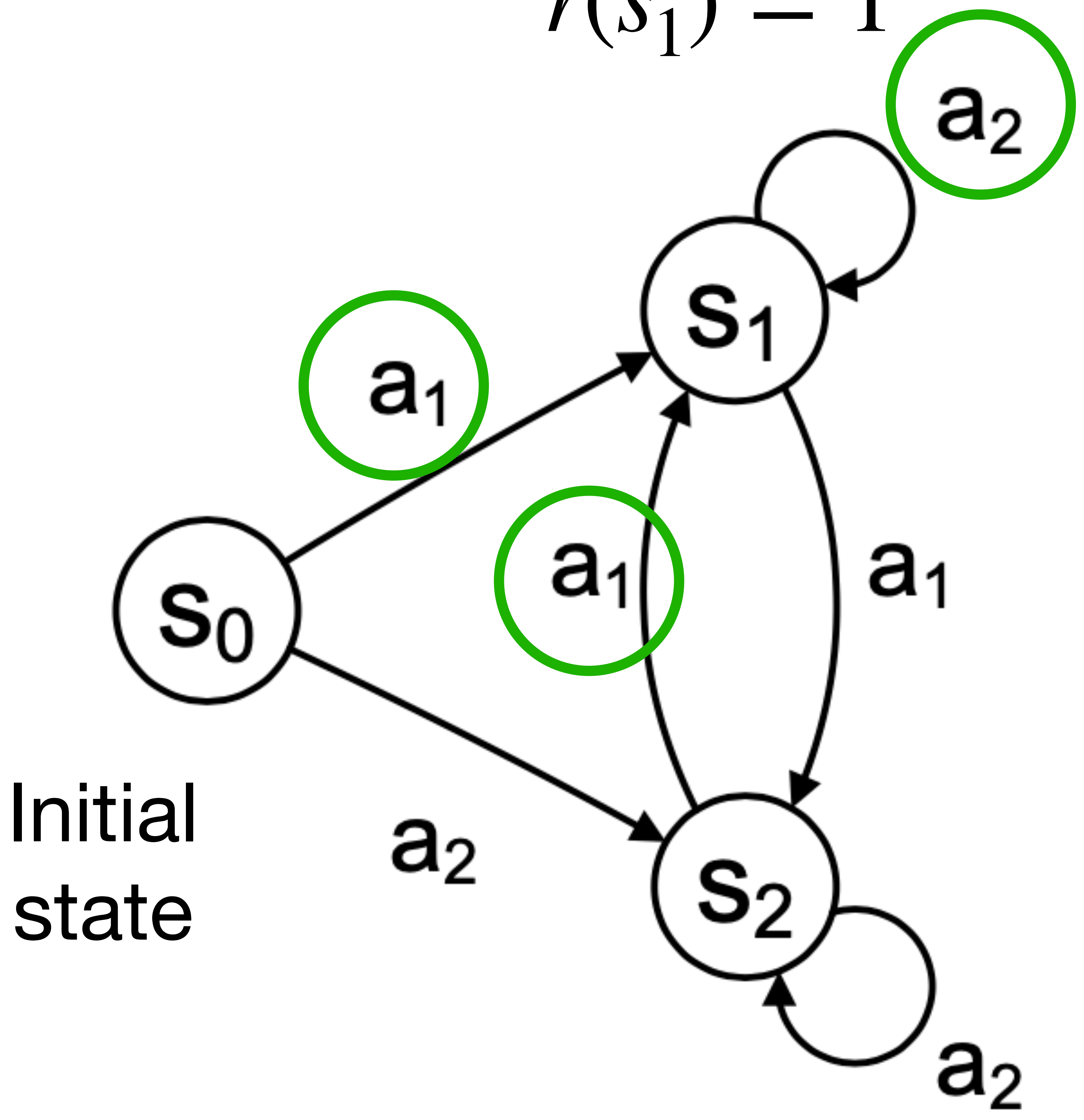
$$\leq H^2\epsilon$$

Distribution Shift Example (H^2 factor is tight)



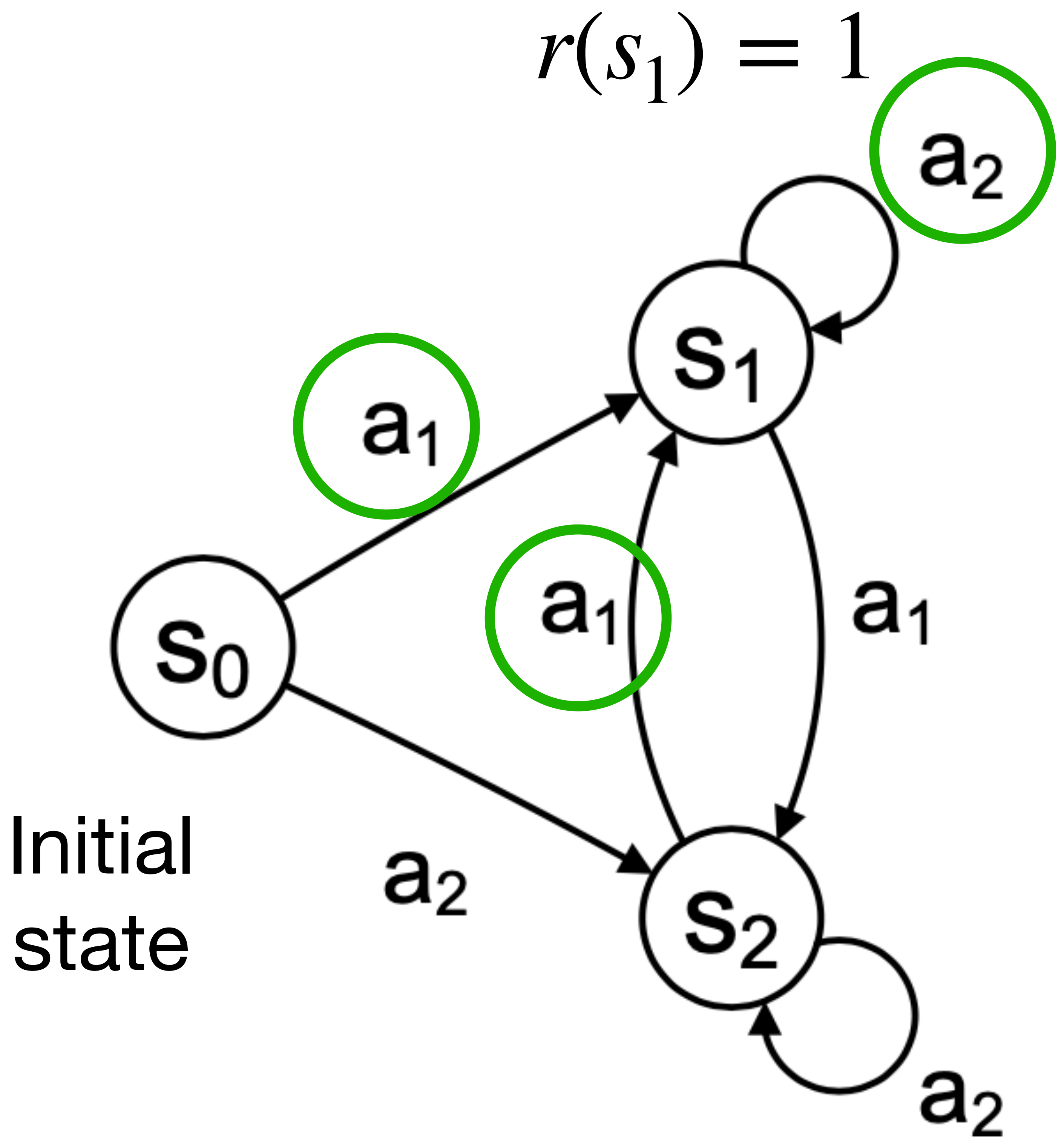
Distribution Shift Example (H^2 factor is tight)

$r(s_1) = 1$



Opt policy:

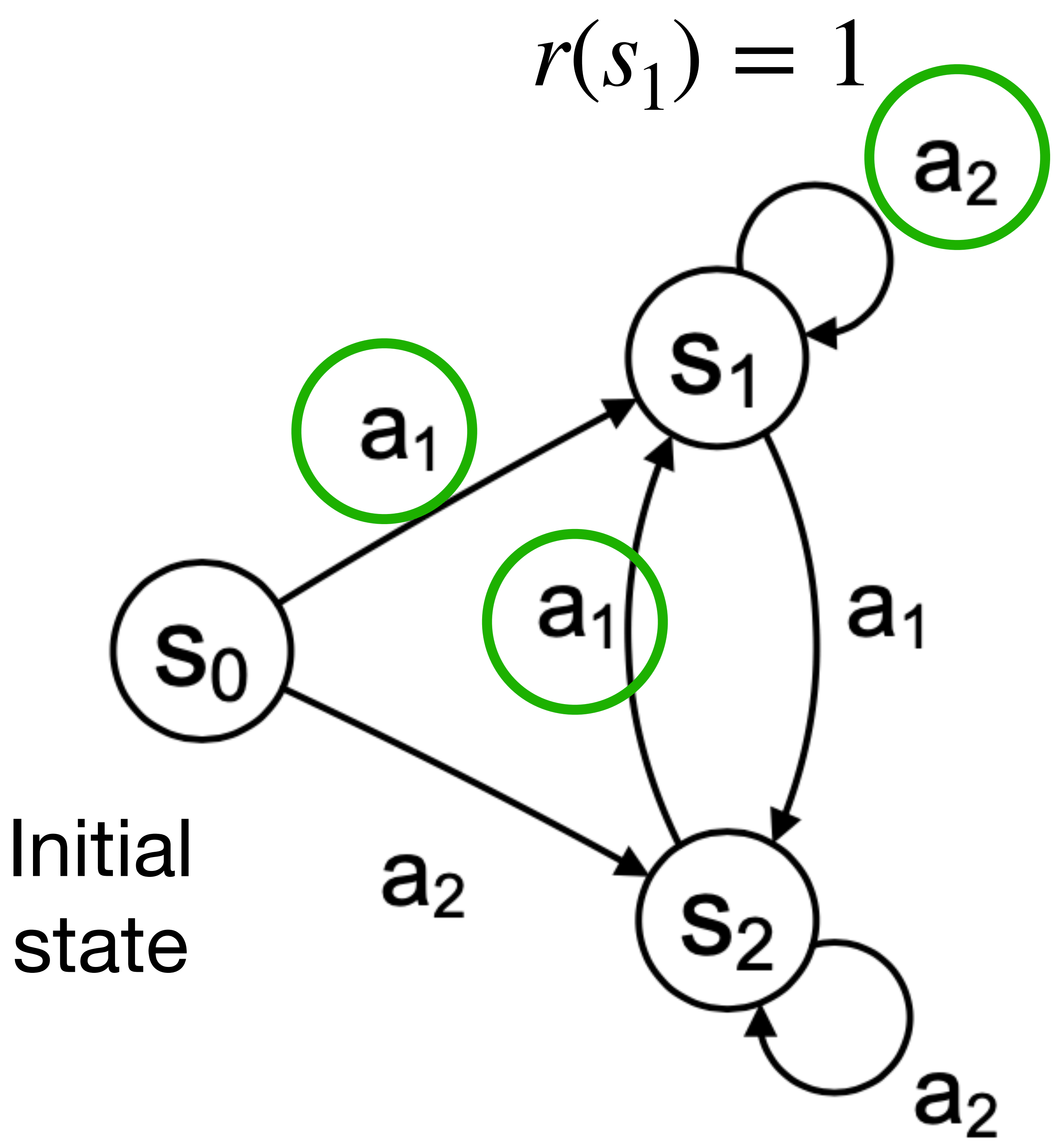
Distribution Shift Example (H^2 factor is tight)



Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

Distribution Shift Example (H^2 factor is tight)



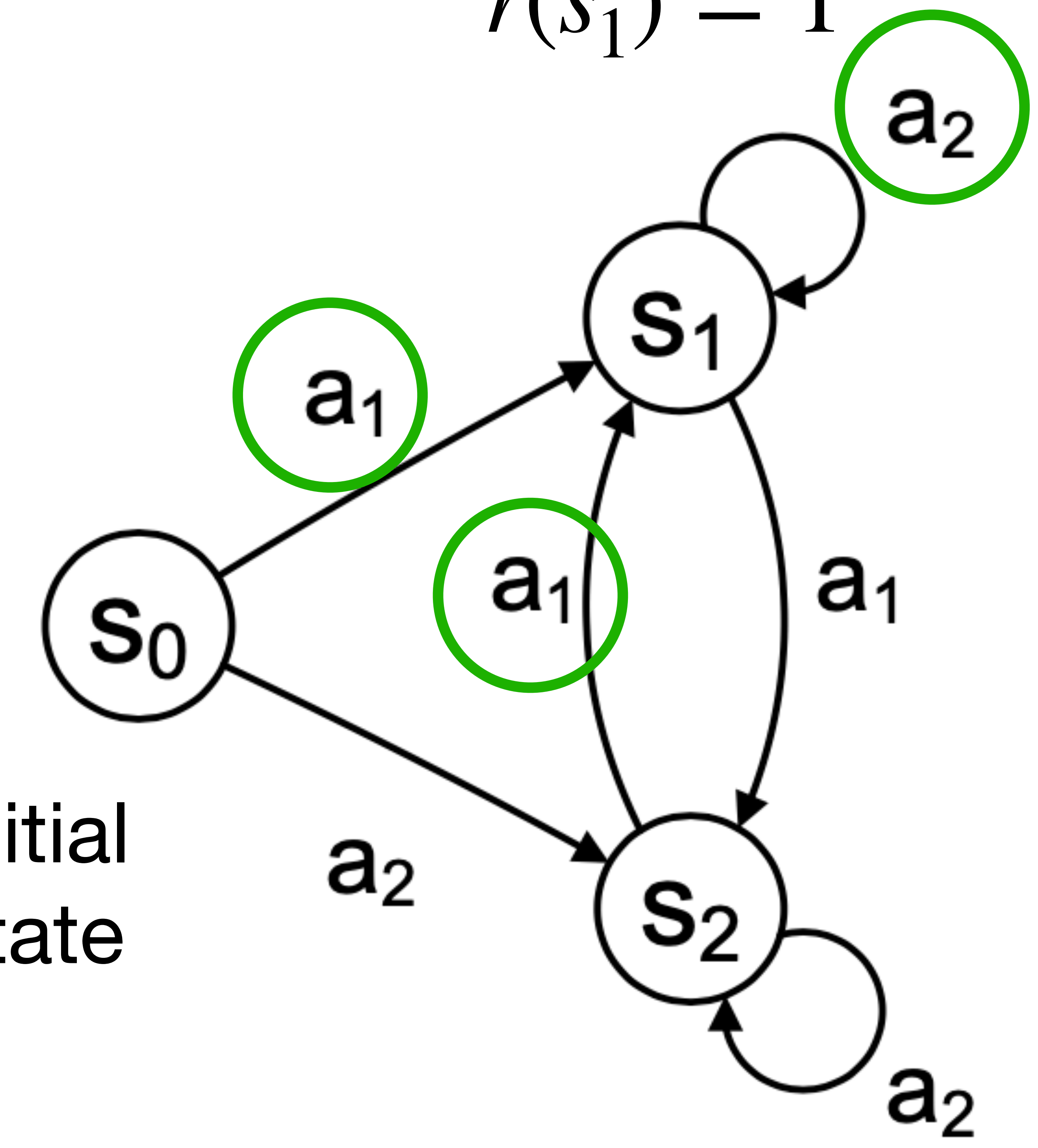
Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

Distribution Shift Example (H^2 factor is tight)

$r(s_1) = 1$



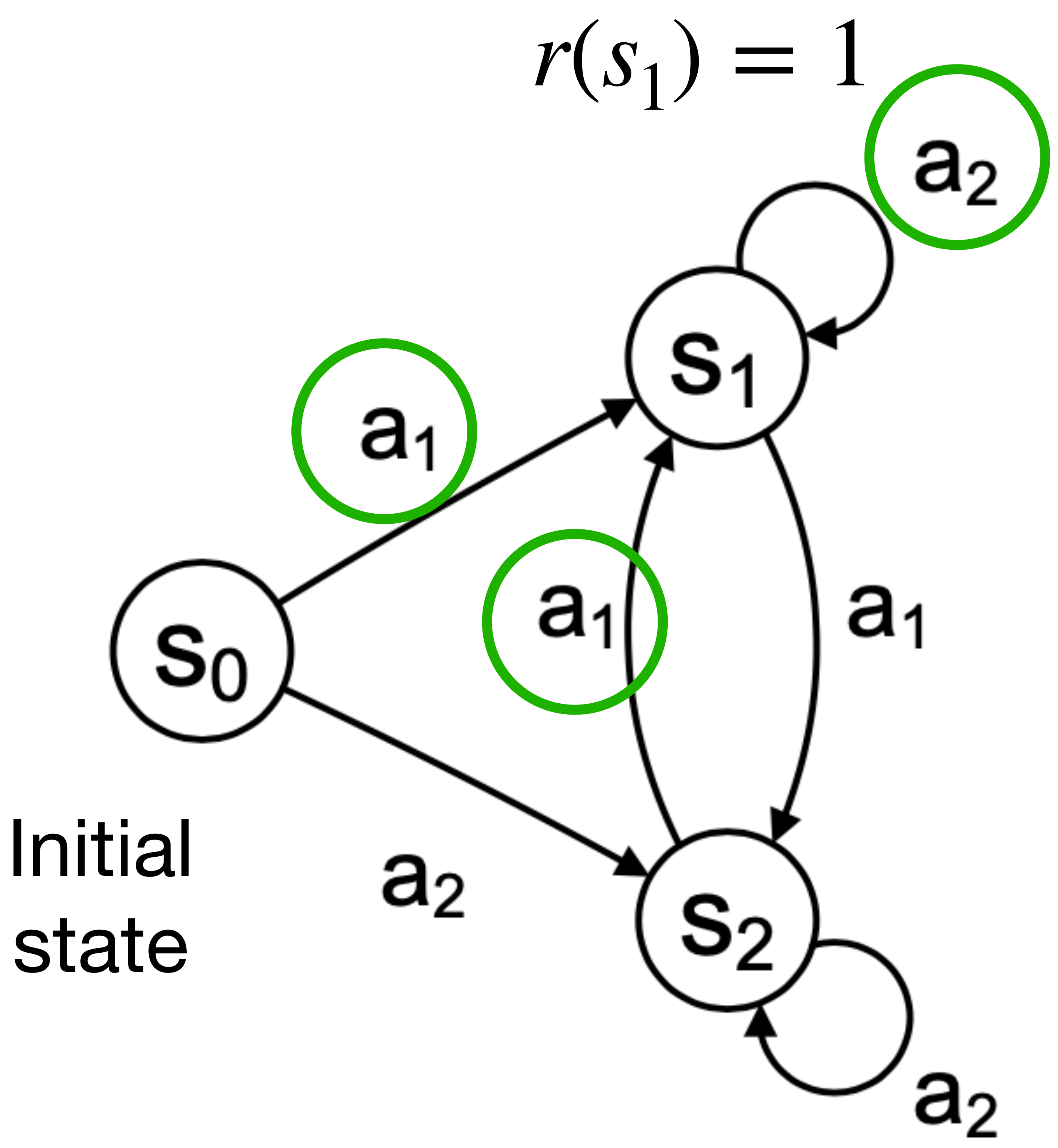
Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

$V_0^{\pi^\star}(s_0) = H - 1$

Distribution Shift Example (H^2 factor is tight)



Assume SL returns the policy $\widehat{\pi}$:

$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - H\epsilon \\ a_2 & \text{w/ prob } H\epsilon \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

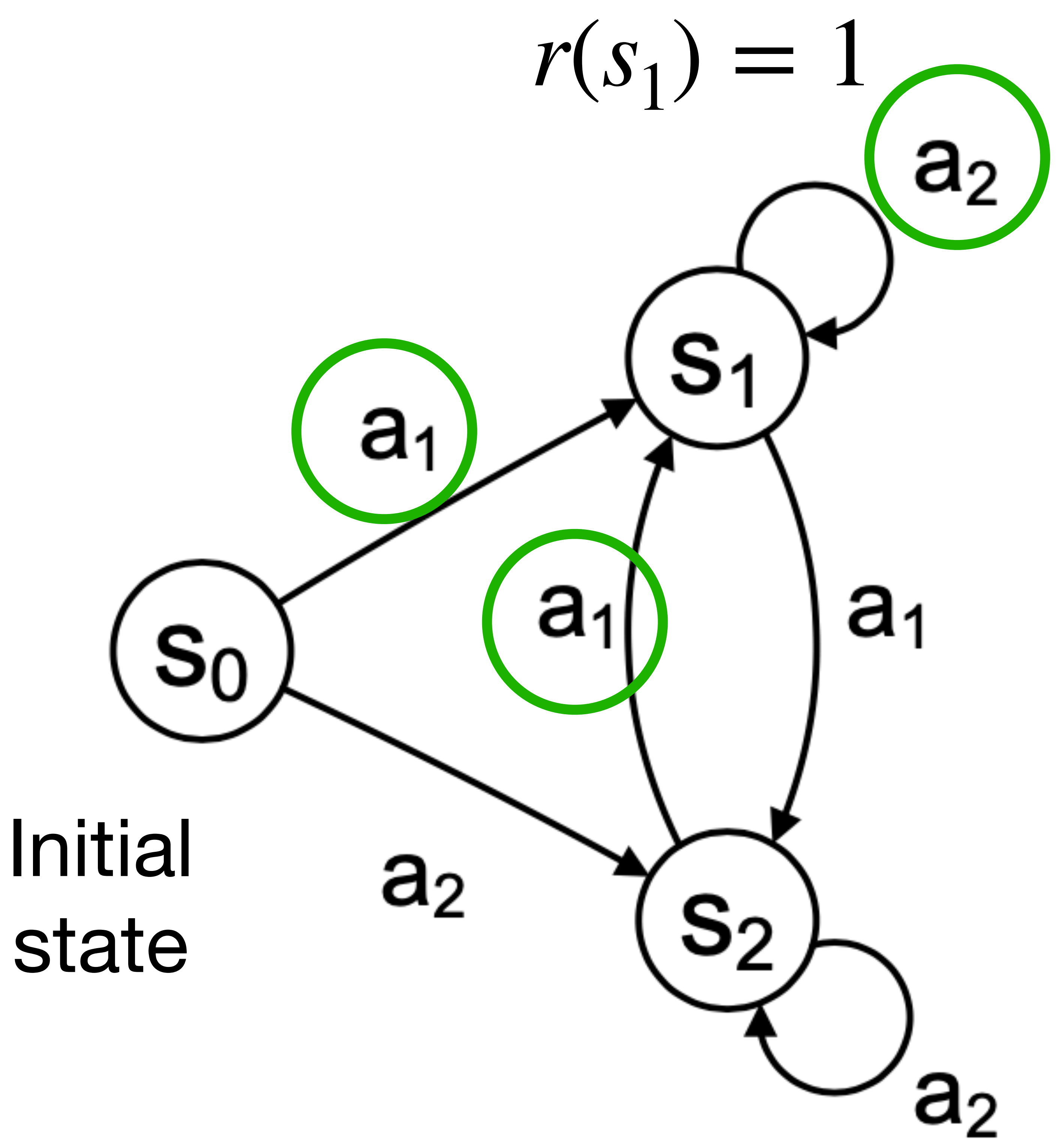
Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

$V_0^{\pi^\star}(s_0) = H - 1$

Distribution Shift Example (H^2 factor is tight)



Assume SL returns the policy $\widehat{\pi}$:

$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - H\epsilon \\ a_2 & \text{w/ prob } H\epsilon \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

This policy has good supervised learning error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1} \left[\widehat{\pi}(s_h) \neq \pi^\star(s_h) \right] \right] = \epsilon$$

note: while $\widehat{\pi}(s_2) \neq \pi^\star(s_2)$, state s_2 is never visited under $\pi^\star$

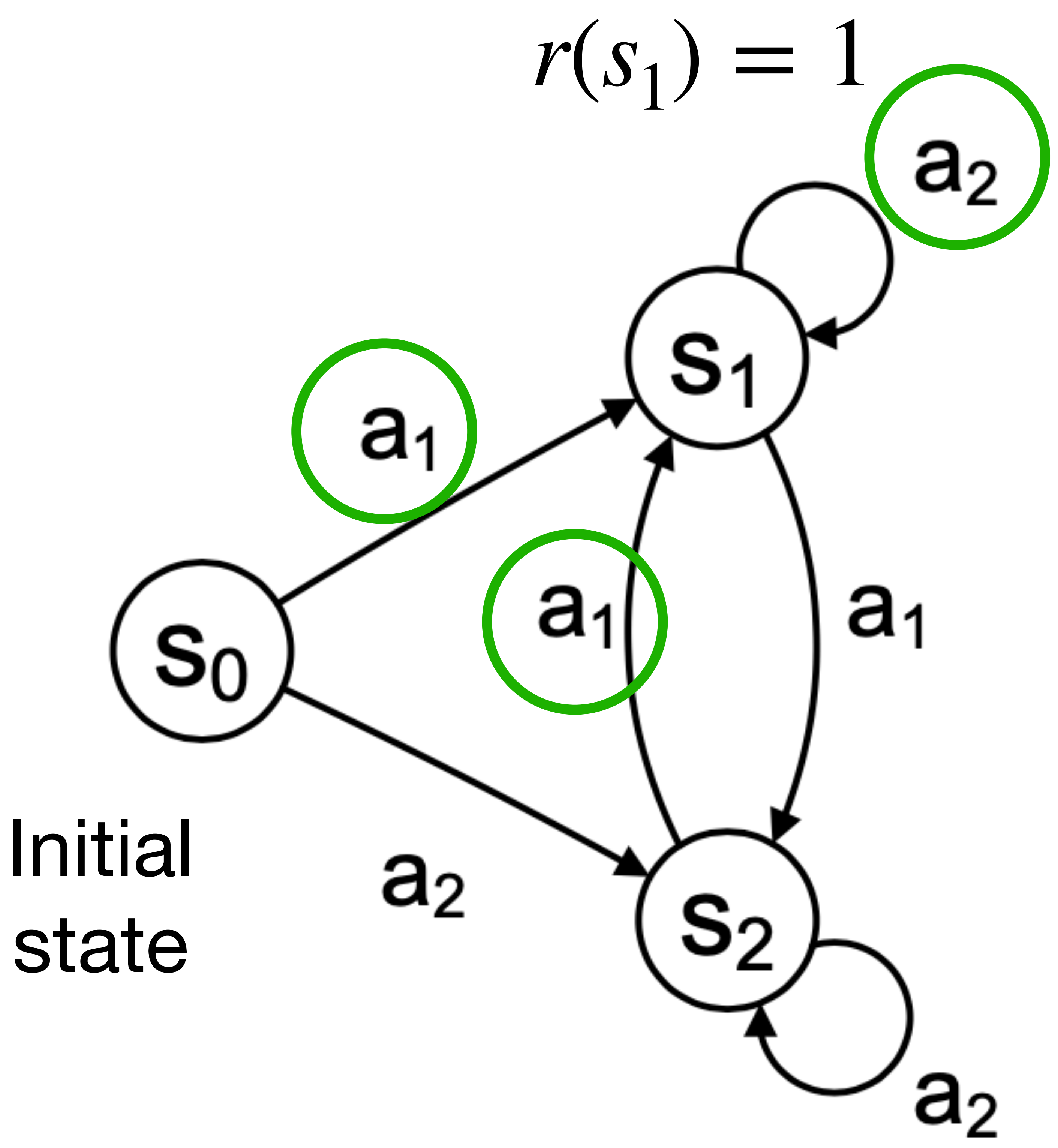
Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

$V_0^{\pi^\star}(s_0) = H - 1$

Distribution Shift Example (H^2 factor is tight)



Assume SL returns the policy $\widehat{\pi}$:

$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - H\epsilon \\ a_2 & \text{w/ prob } H\epsilon \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

This policy has good supervised learning error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1} \left[\widehat{\pi}(s_h) \neq \pi^\star(s_h) \right] \right] = \epsilon$$

note: while $\widehat{\pi}(s_2) \neq \pi^\star(s_2)$, state s_2 is never visited under $\pi^\star$

We have quadratic degradation (in H):

$$V_0^{\widehat{\pi}}(s_0) = (1 - H\epsilon) \cdot V_0^{\pi^\star}(s_0) + H\epsilon \cdot 0 = V_0^{\pi^\star}(s_0) - \epsilon H(H-1)$$

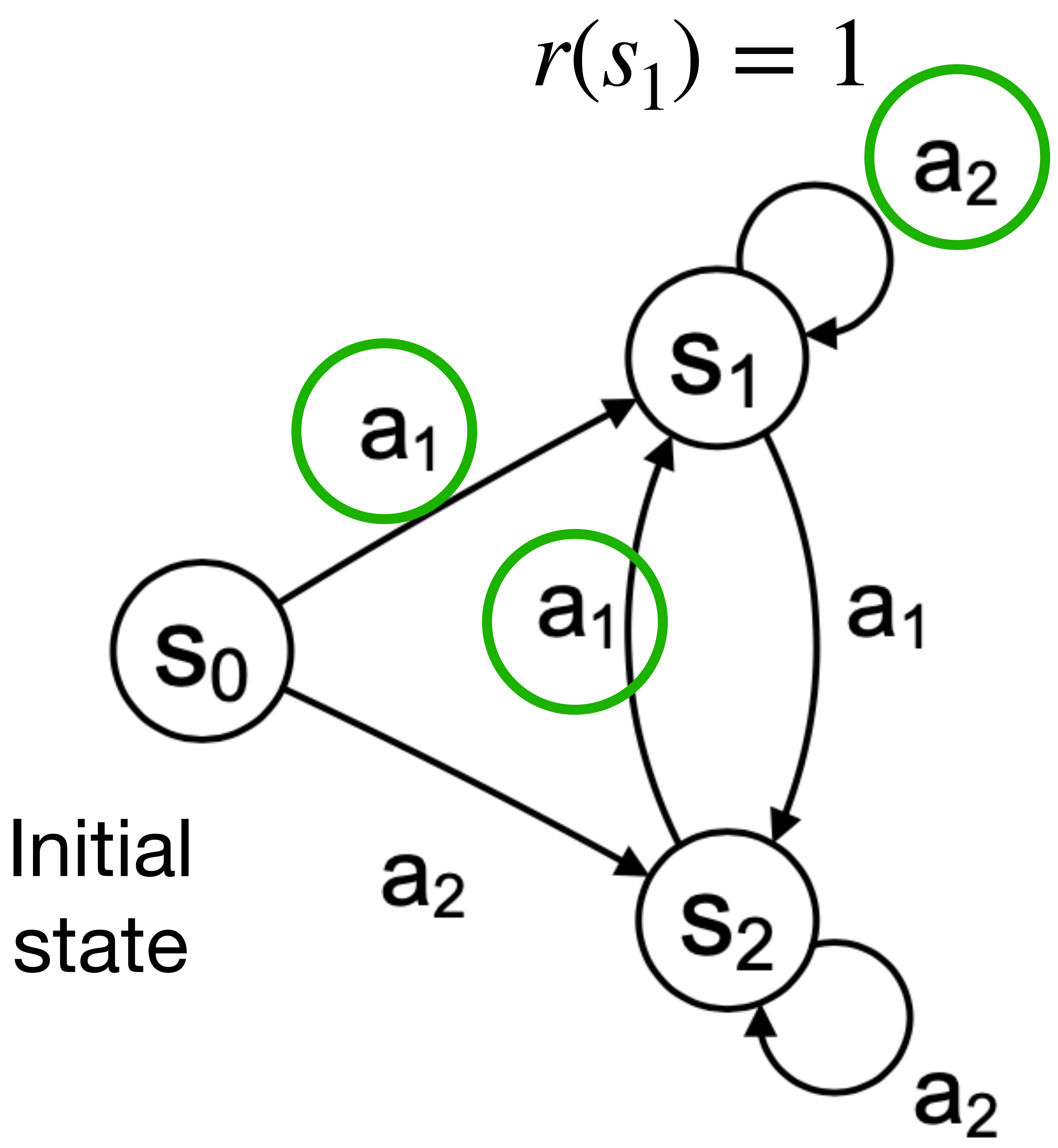
Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

$V_0^{\pi^\star}(s_0) = H - 1$

Distribution Shift Example (H^2 factor is tight)



Opt policy:

Under $\rho_{\pi^\star}$, trajectory is $s_0, s_1, s_1, \ldots$

$\rho_{\pi^\star}(s_h = s_2) = 0$

$V_0^{\pi^\star}(s_0) = H - 1$

Assume SL returns the policy $\widehat{\pi}$:

$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - H\epsilon \\ a_2 & \text{w/ prob } H\epsilon \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

This policy has good supervised learning error:

$$\mathbb{E}_{\tau \sim \rho_{\pi^\star}} \left[\frac{1}{H} \sum_{h=0}^{H-1} \mathbf{1}\left[\widehat{\pi}(s_h) \neq \pi^\star(s_h) \right] \right] = \epsilon$$

note: while $\widehat{\pi}(s_2) \neq \pi^\star(s_2)$, state s_2 is never visited under $\pi^\star$

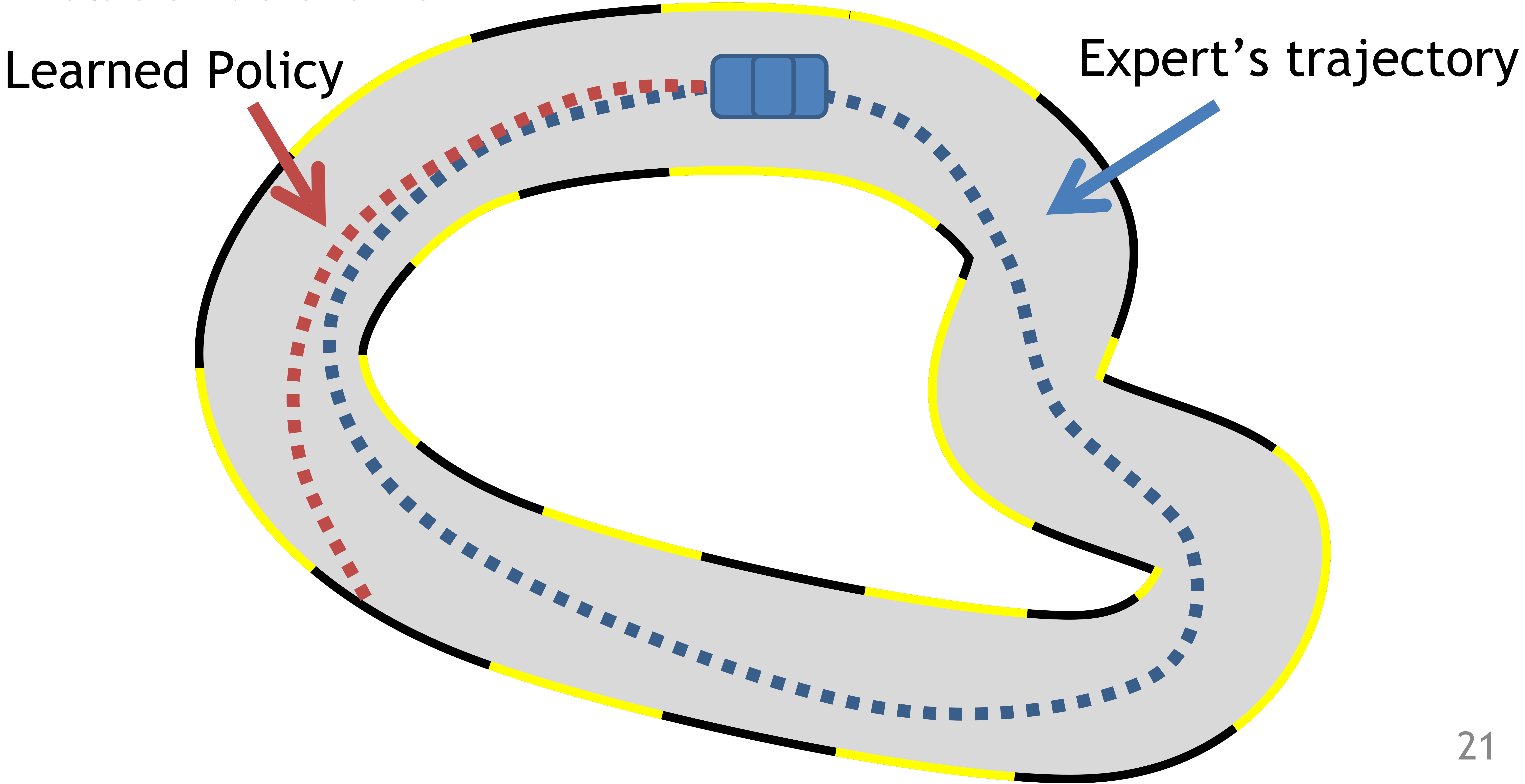
We have quadratic degradation (in H):

$$V_0^{\widehat{\pi}}(s_0) = (1 - H\epsilon) \cdot V_0^{\pi^\star}(s_0) + H\epsilon \cdot 0 = V_0^{\pi^\star}(s_0) - \epsilon H(H-1)$$

Intuition: once we make a mistake at s_0, we end up in s_2 which is not in the training data!

What could go wrong?

- Predictions affect future inputs/ observations



Expert Demos


Ready!


Ready!

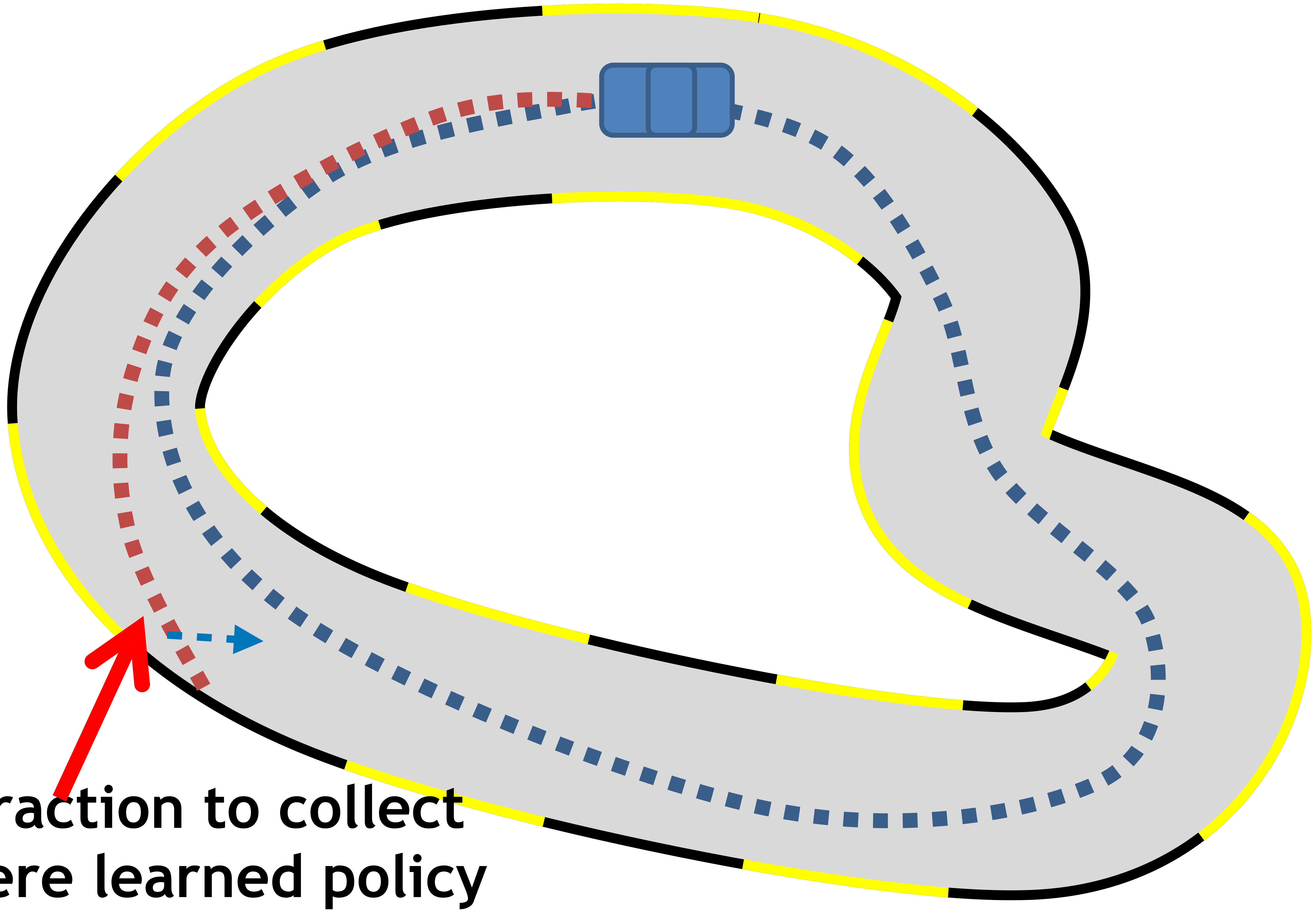

Features
Ready!


Features
Ready!

Today

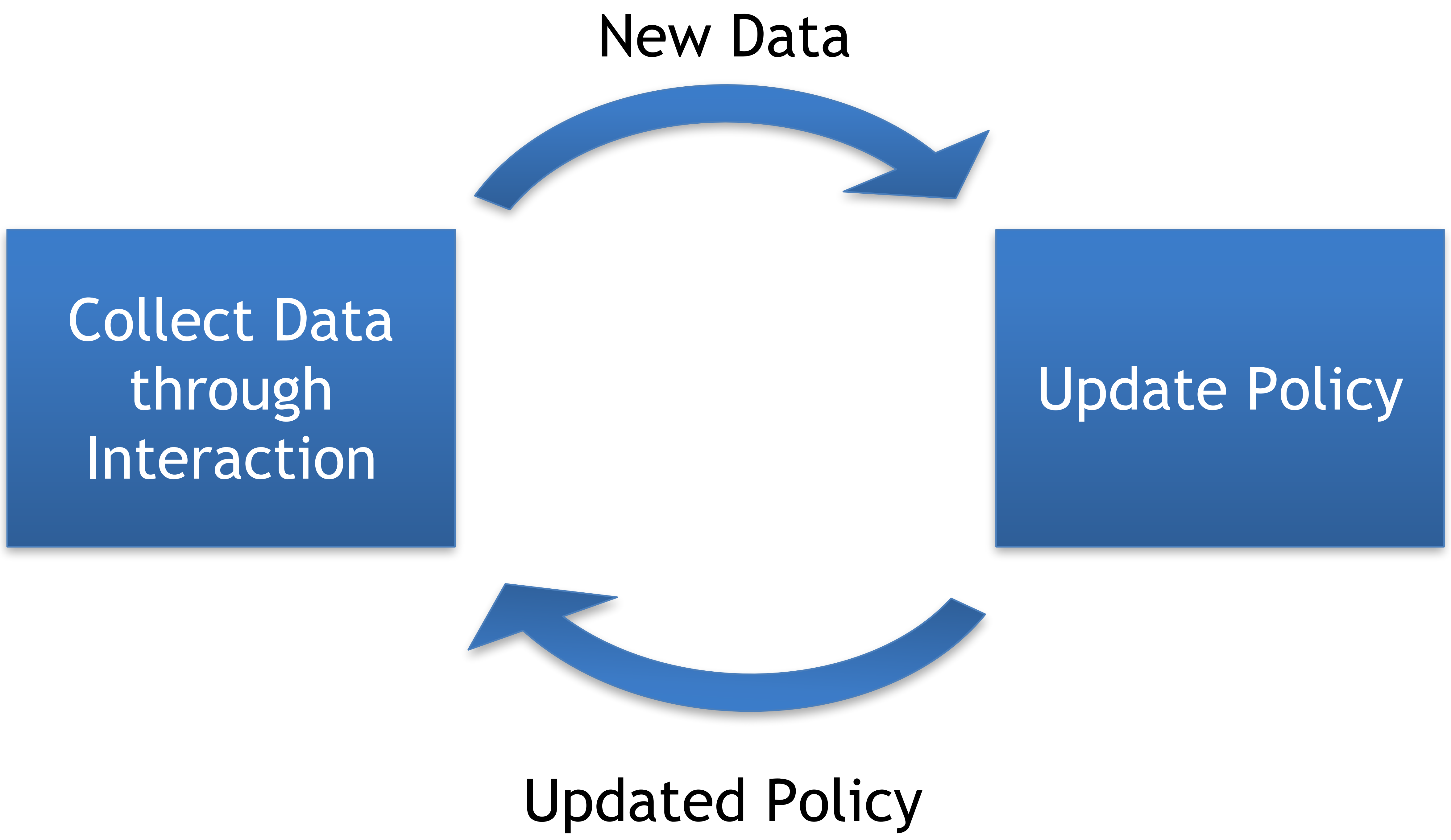
- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

Intuitive solution: **Interaction**



Use interaction to collect data where learned policy goes

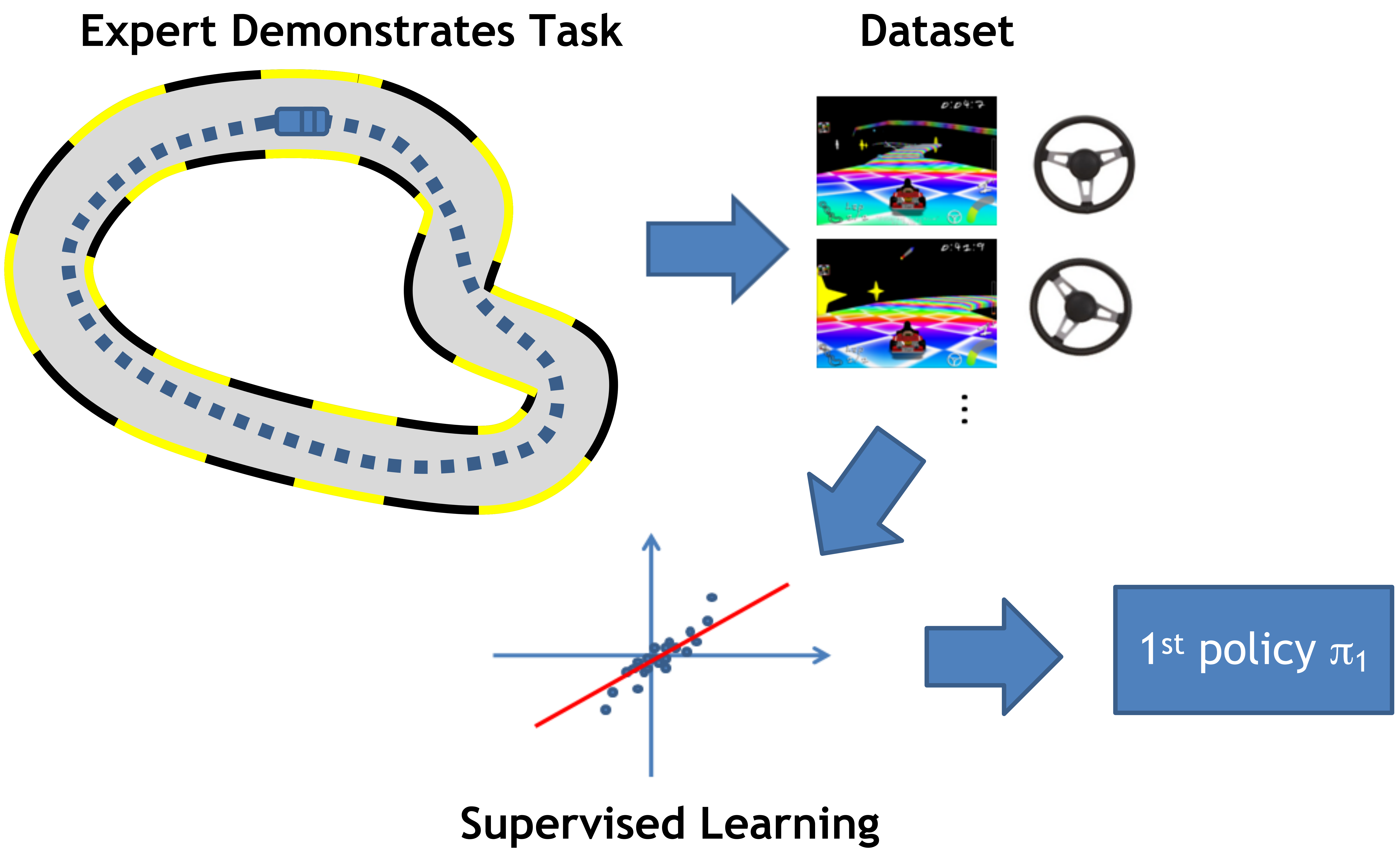
General Idea: Iterative Interactive Approach



DAgger: Dataset Aggregation

[Ross11a]

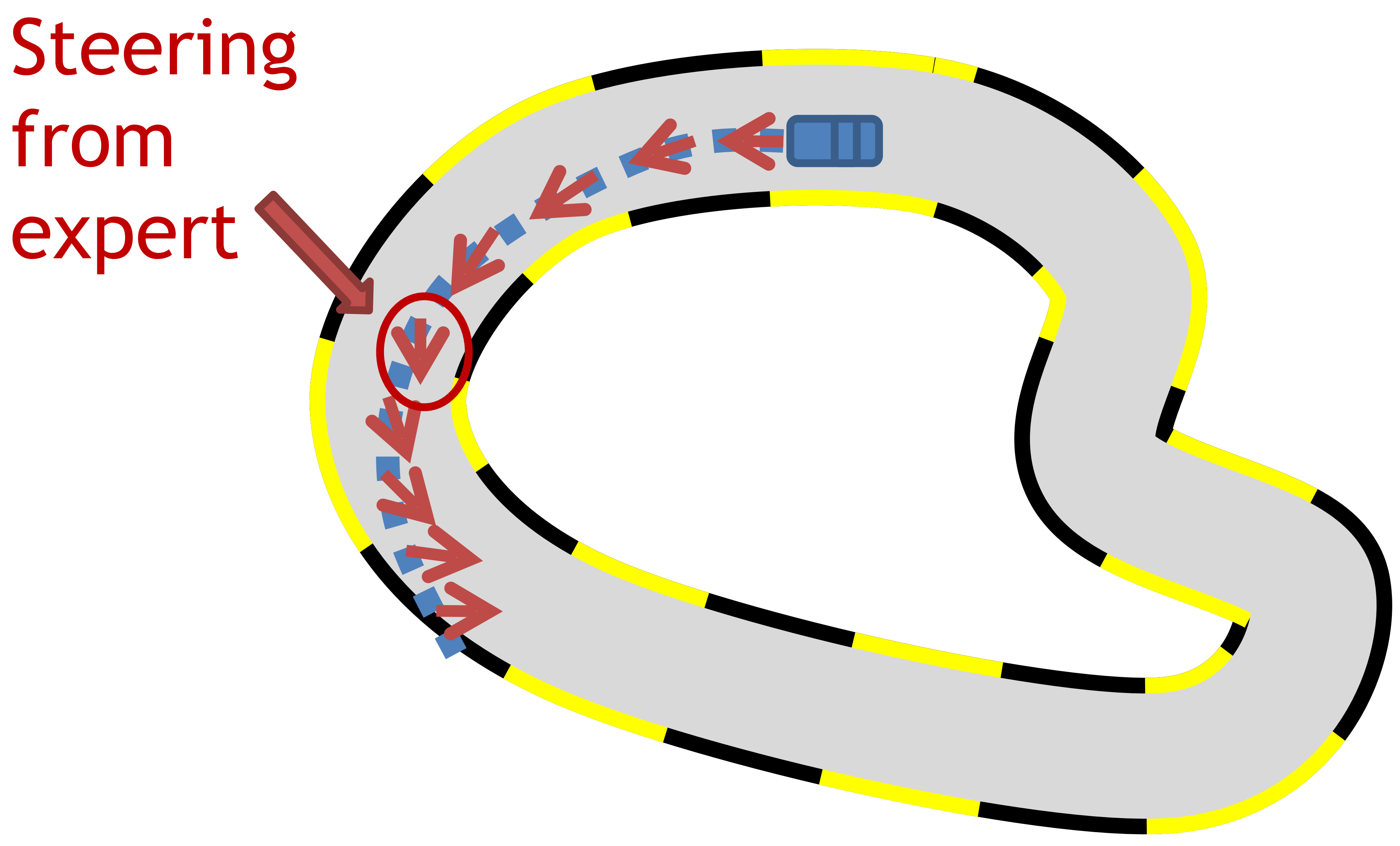
0th iteration



DAgger: Dataset Aggregation [Ross11a]

1st iteration

Execute π_1 and Query Expert

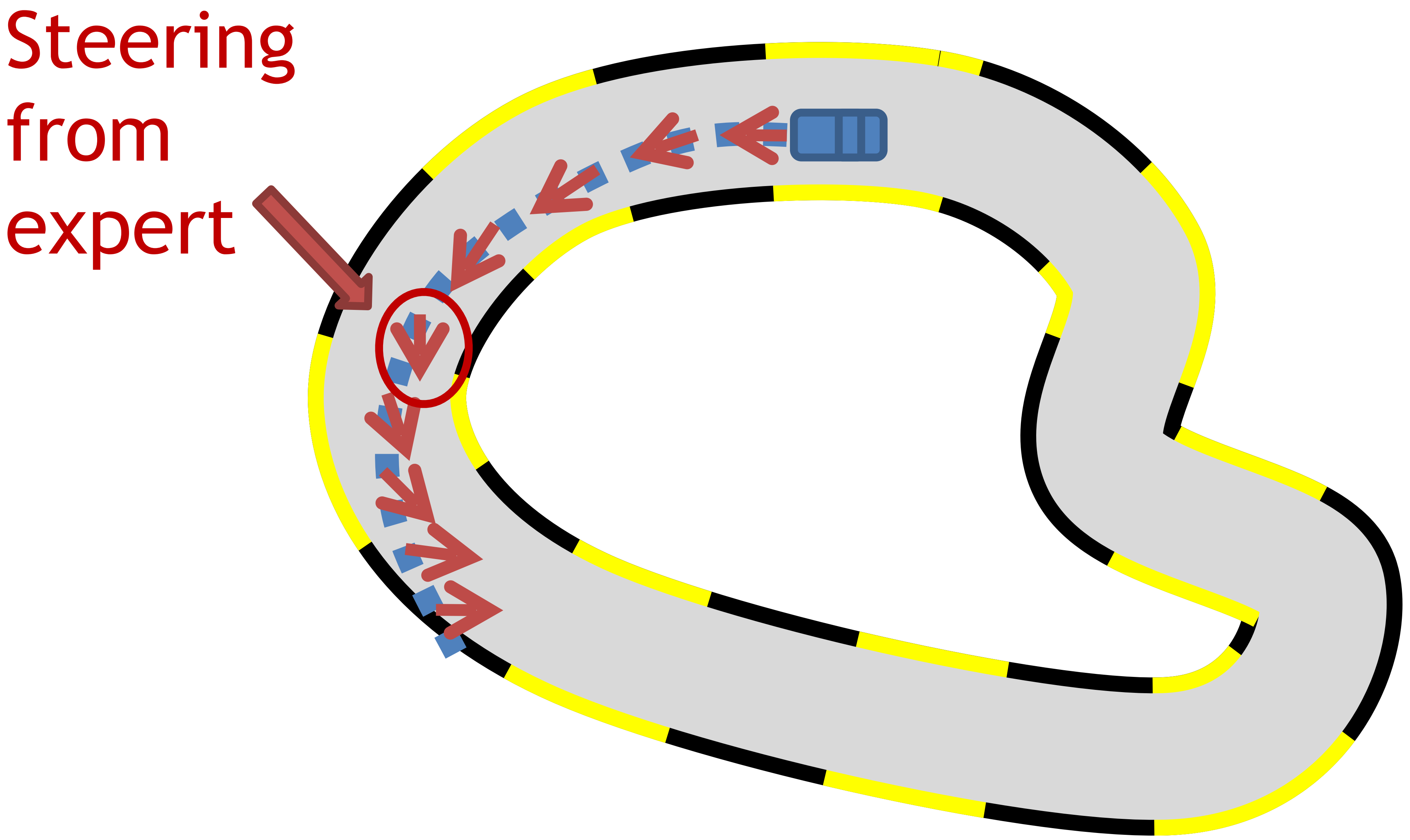


DAgger: Dataset Aggregation

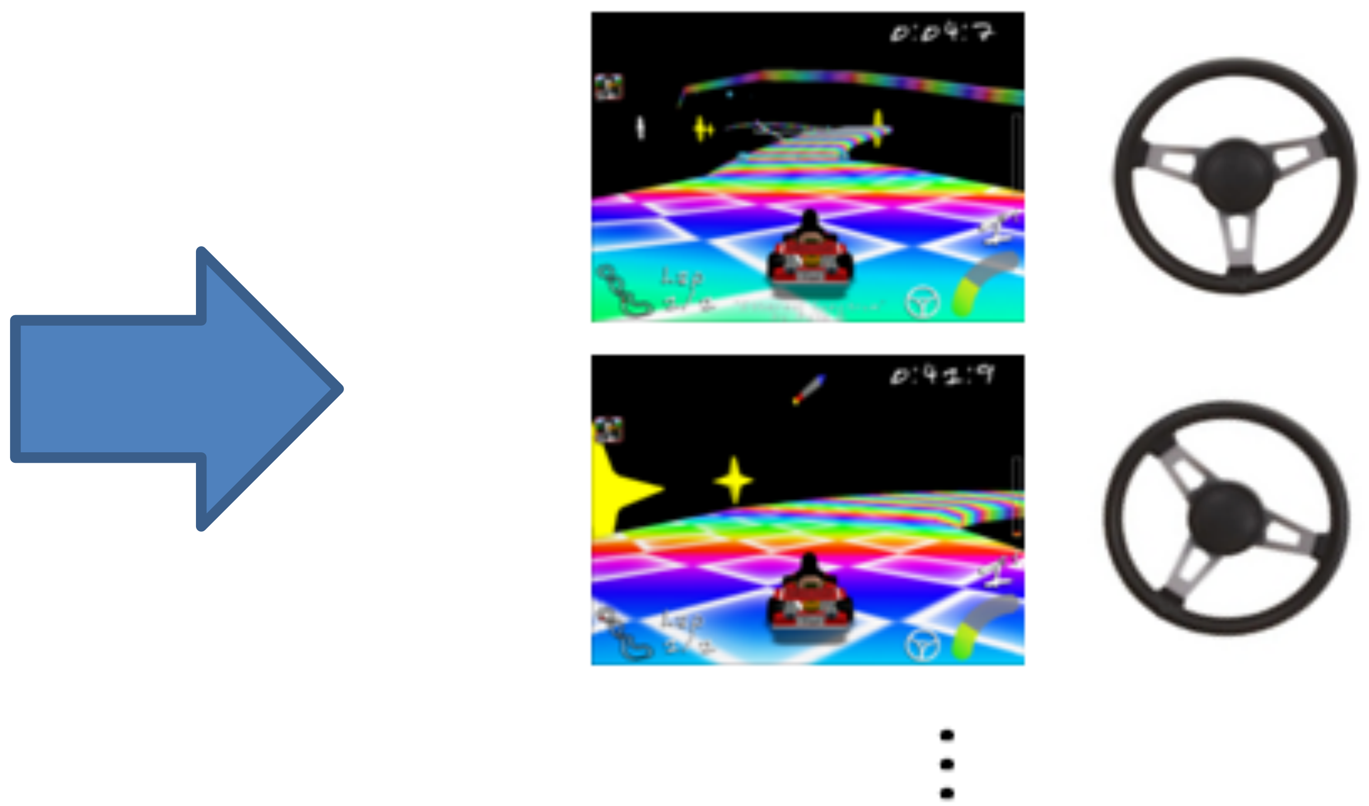
[Ross11a]

1st iteration

Execute π_1 and Query Expert



New Data



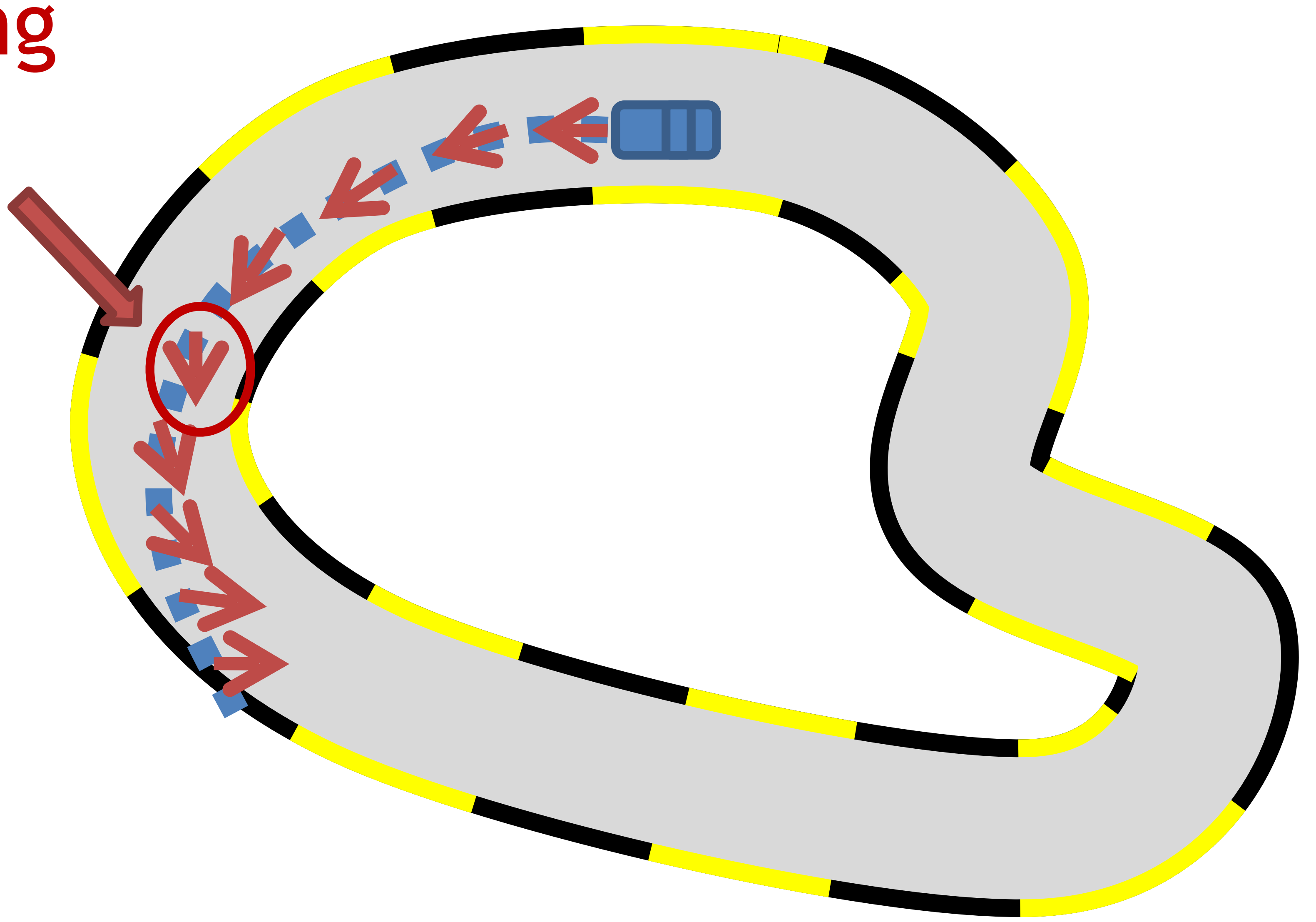
DAgger: Dataset Aggregation

[Ross11a]

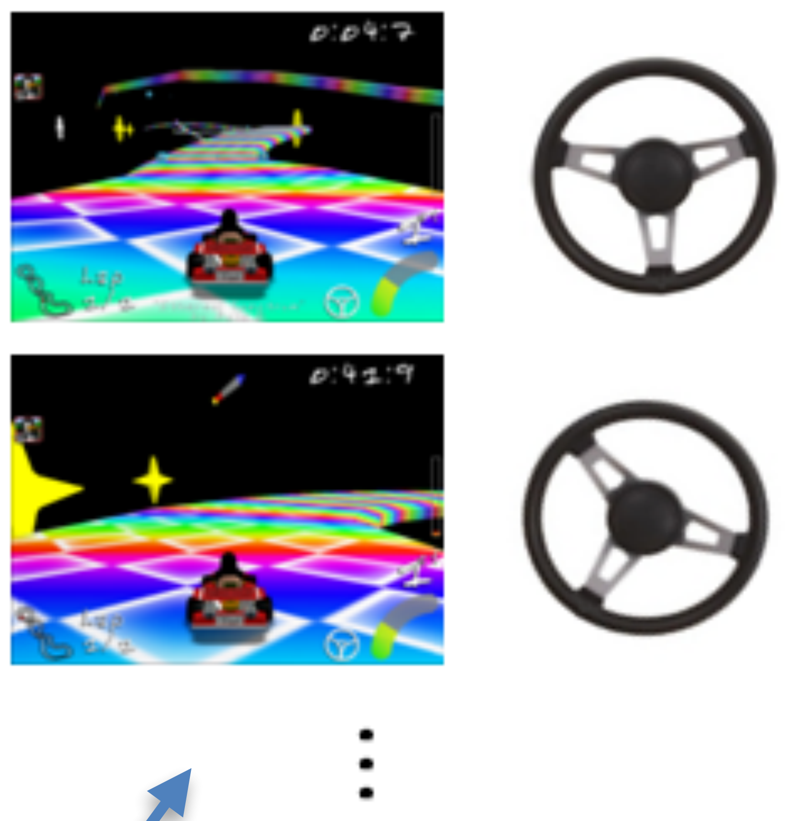
1st iteration

Execute π_1 and Query Expert

Steering
from
expert



New Data



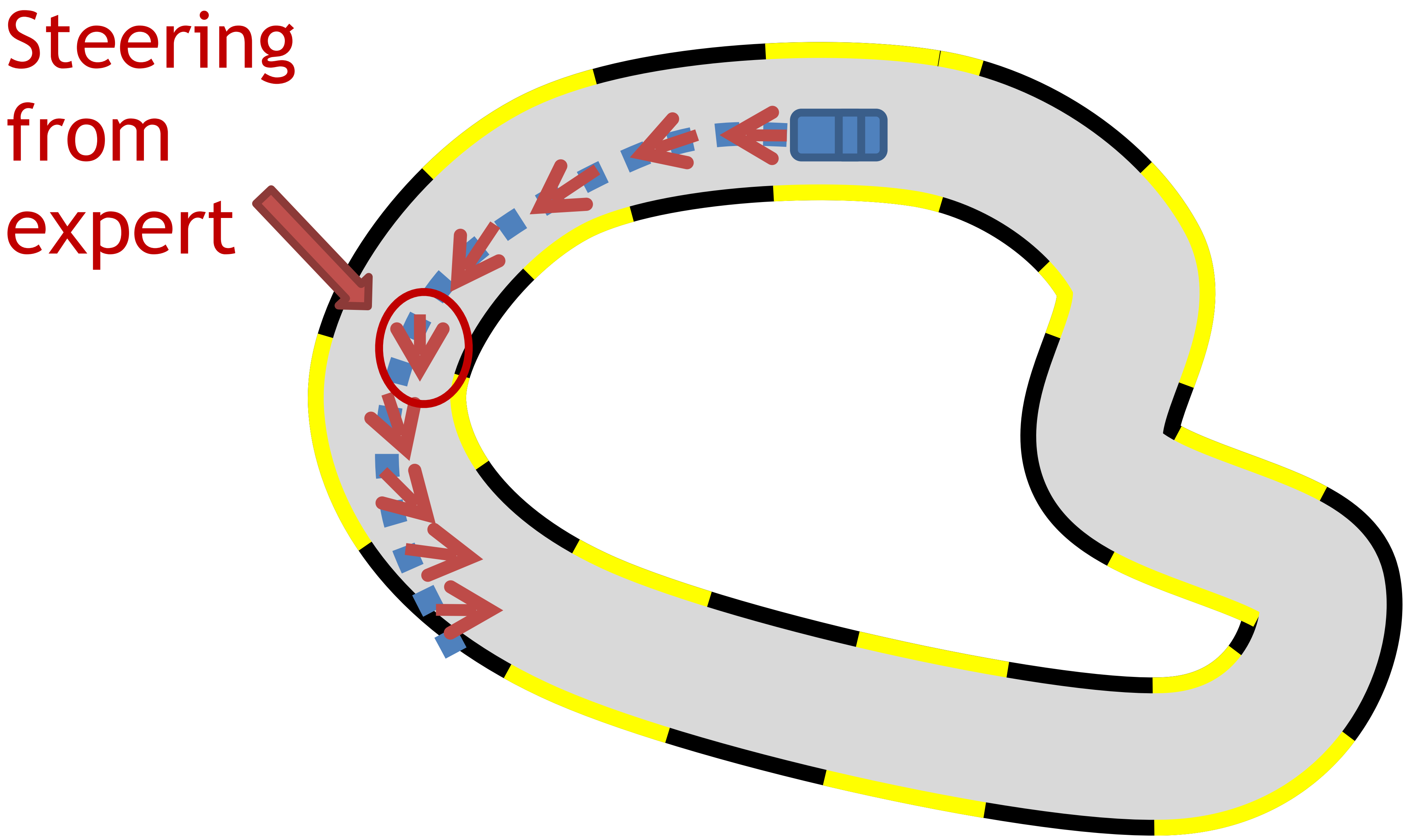
States from
the learned policy

DAgger: Dataset Aggregation

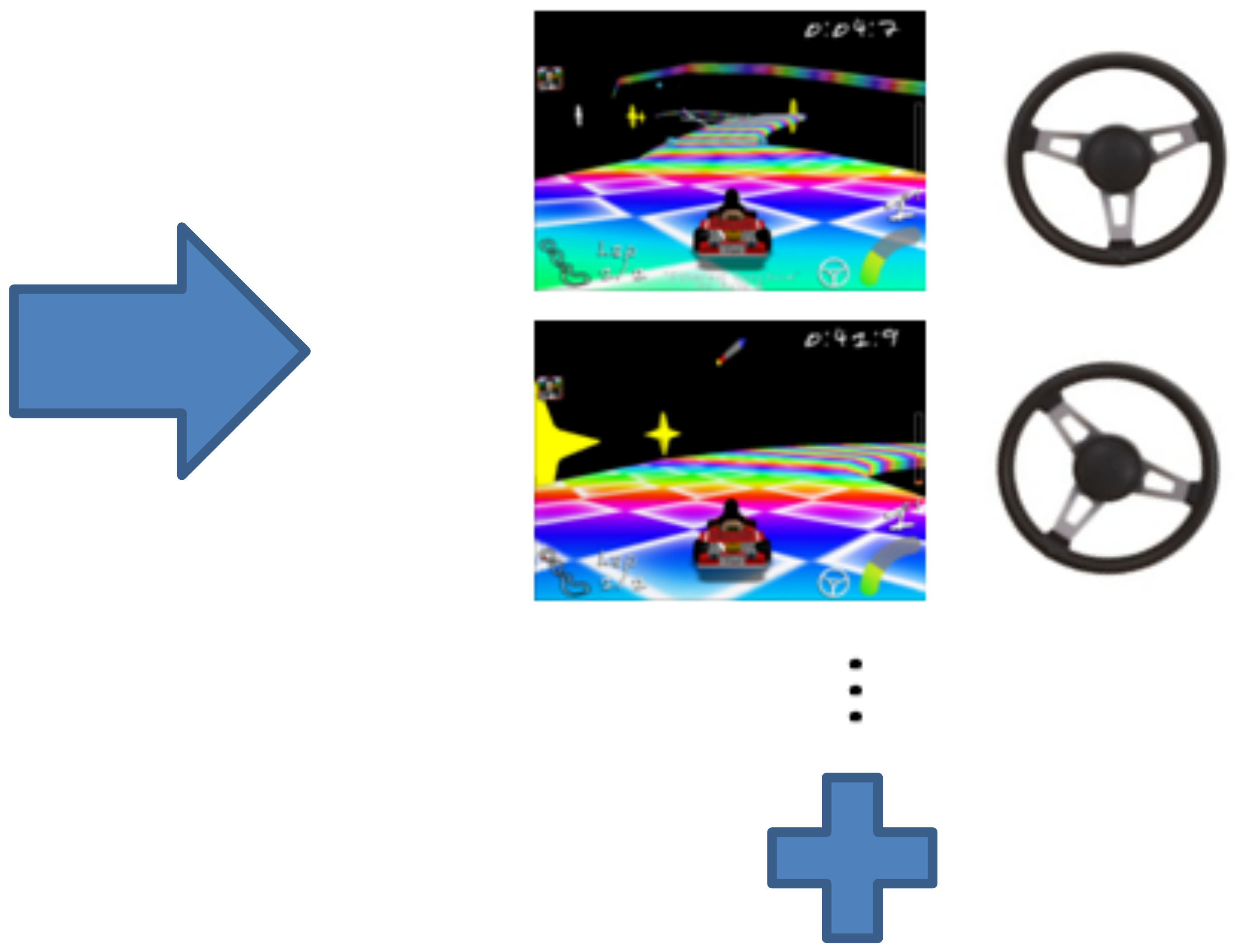
[Ross11a]

1st iteration

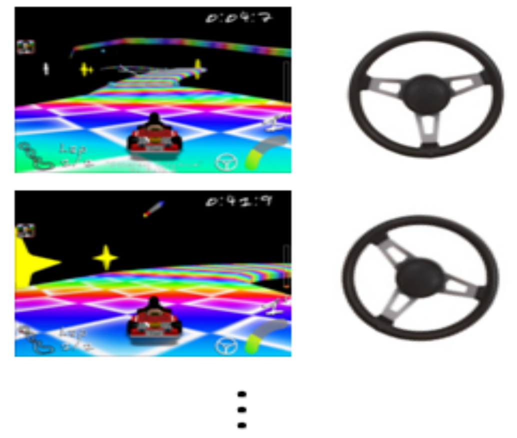
Execute π_1 and Query Expert



New Data



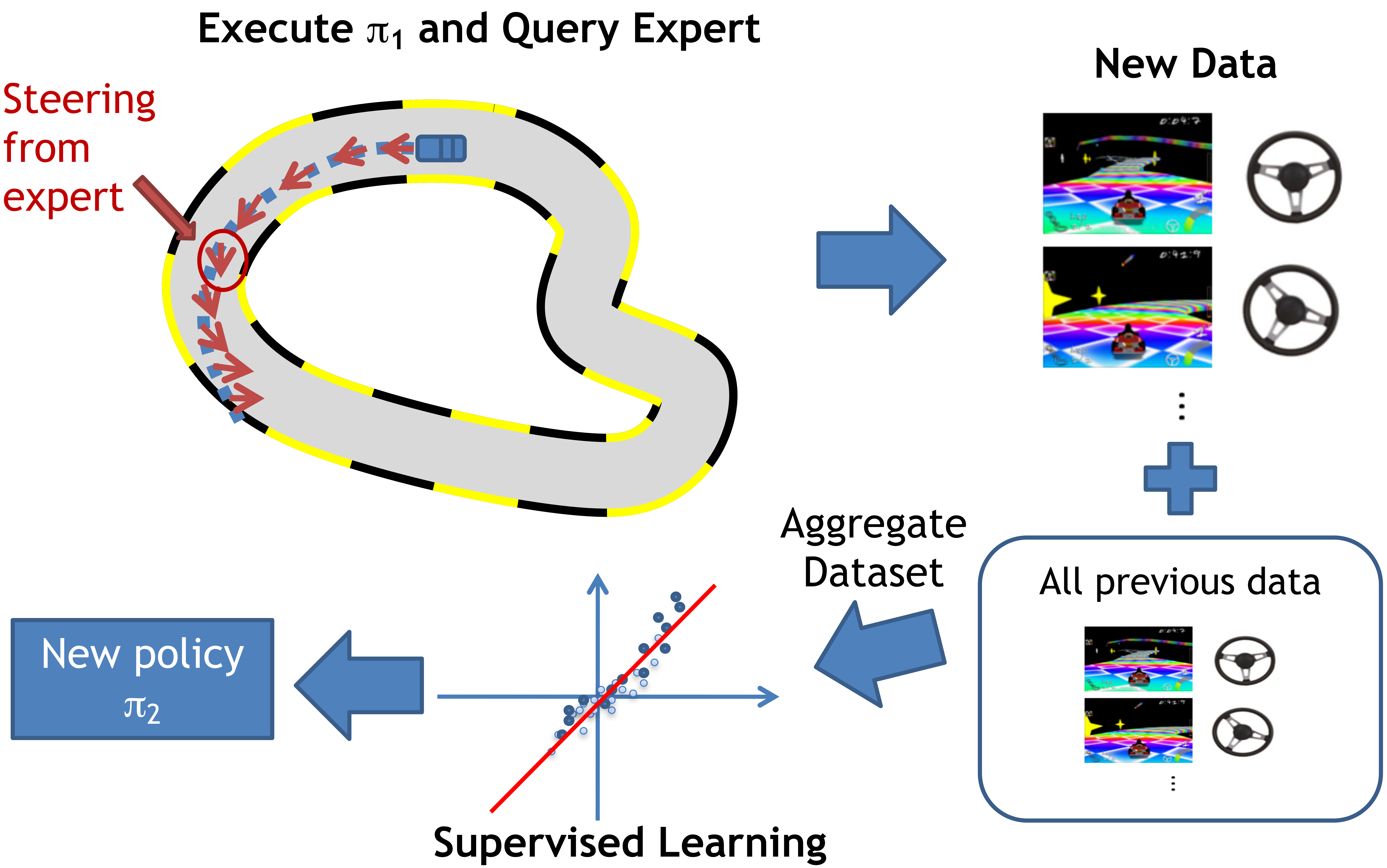
All previous data



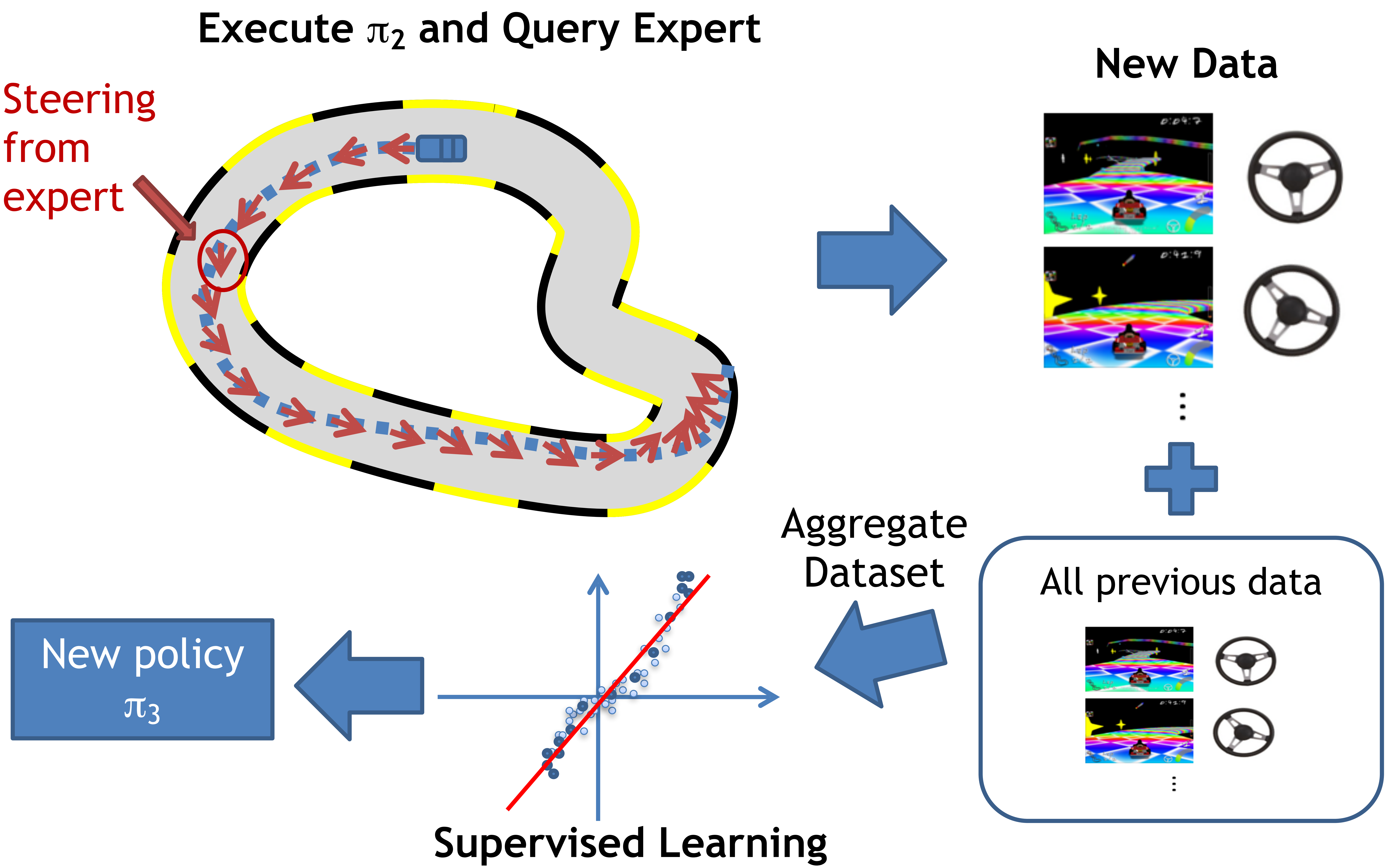
DAgger: Dataset Aggregation

[Ross11a]

1st iteration

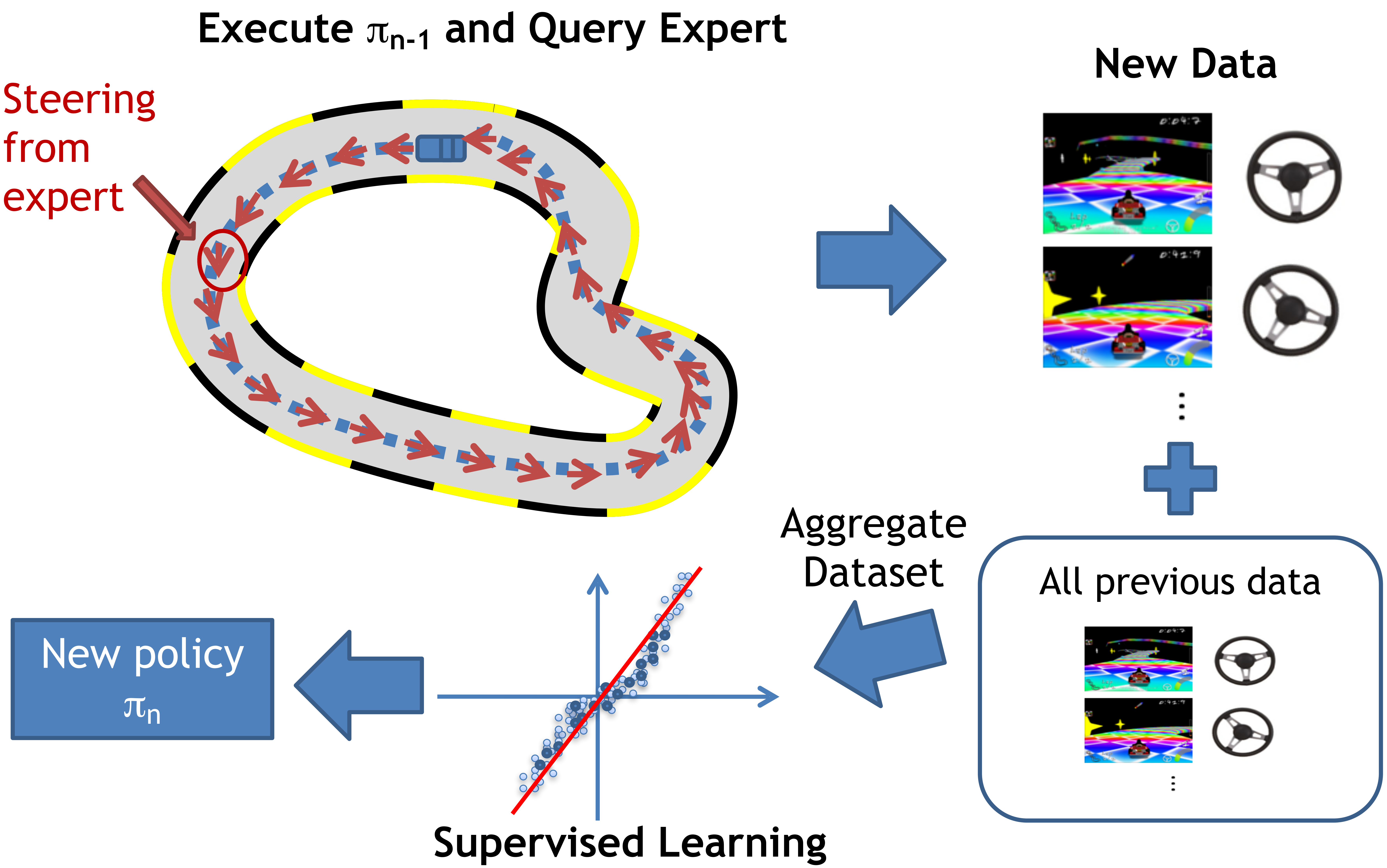


DAgger: Dataset Aggregation [Ross11a]
2nd iteration



DAgger: Dataset Aggregation [Ross11a]

n^{th} iteration



The DAgger algorithm

Initialize π^0, and dataset $\mathscr{D} = \emptyset$

For $t = 0 \rightarrow T - 1$:

The DAgger algorithm

Initialize π^0, and dataset $\mathscr{D} = \emptyset$

For $t = 0 \to T - 1$:

1. W/ π^t, generate dataset of trajectories $\mathscr{D}^t = \{\tau_1, \tau_2, \ldots\}$
where for all trajectories $s_h \sim \rho_{\pi^t}$, $a_h = \pi^\star(s_h)$

The DAgger algorithm

Initialize π^0, and dataset $\mathscr{D} = \emptyset$

For $t = 0 \rightarrow T - 1$:

1. W/ π^t, generate dataset of trajectories $\mathscr{D}^t = \{\tau_1, \tau_2, \ldots\}$
where for all trajectories $s_h \sim \rho_{\pi^t}, \; a_h = \pi^\star(s_h)$
2. **Data aggregation:** $\mathscr{D} = \mathscr{D} \cup \mathscr{D}^t$

The DAgger algorithm

Initialize π^0, and dataset $\mathscr{D} = \emptyset$

For $t = 0 \to T - 1$:

1. W/ π^t, generate dataset of trajectories $\mathscr{D}^t = \{\tau_1, \tau_2, \ldots\}$ where for all trajectories $s_h \sim \rho_{\pi^t}$, $a_h = \pi^\star(s_h)$
2. **Data aggregation:** $\mathscr{D} = \mathscr{D} \cup \mathscr{D}^t$
3. **Update policy via Supervised-Learning**: $\pi^{t+1} = \text{SL}\left(\mathscr{D}\right)$

The DAgger algorithm

Initialize π^0, and dataset $\mathscr{D} = \varnothing$

For $t = 0 \rightarrow T - 1$:

1. W/ π^t, generate dataset of trajectories $\mathscr{D}^t = \{\tau_1, \tau_2, \ldots\}$
where for all trajectories $s_h \sim \rho_{\pi^t}$, $a_h = \pi^\star(s_h)$
2. **Data aggregation:** $\mathscr{D} = \mathscr{D} \cup \mathscr{D}^t$
3. **Update policy via Supervised-Learning**: $\pi^{t+1} = \text{SL}\left(\mathscr{D}\right)$

In practice, the DAgger algorithm requires less human labeled data than BC.

[Informal Theorem] Under more assumptions + assuming ϵ SL error is achievable,
the DAgger algorithm has error: $|V^{\pi^\star} - V^{\widehat{\pi}}| \leq H\epsilon$

Success!

[Ross AISTATS 2011]

Success!

[Ross AISTATS 2011]

Today

- Feedback from last lecture
- Recap
- Imitation Learning problem statement
- Behavioral Cloning
- DAgger

Summary:

1. IL can help a lot to explore the space
2. BC pretty good but brittle -> quadratic-in-horizon error
3. Online expert feedback can help with robustness -> linear-in-horizon error

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy