Introduction to Imitation Learning & the Behavior Cloning Algorithm

Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$

State visitation: $d_\mu^\pi(s) = (1-\gamma)\sum_{h=0}^{\infty}\gamma^h \mathbb{P}_h^\pi(s; s_0)$

Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$

State visitation: $d_\mu^\pi(s) = (1-\gamma)\sum_{h=0}^{\infty} \gamma^h \mathbb{P}_h^\pi(s; s_0)$

Performance Difference Lemma:

What's the perf difference between π & π'?

Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$

State visitation: $d^{\pi}_{\mu}(s) = (1-\gamma)\sum_{h=0}^{\infty}\gamma^h \mathbb{P}^{\pi}_h(s; s_0)$

Performance Difference Lemma:

What's the perf difference between π & π'?

$$V^{\pi}_{\mu} - V^{\pi'}_{\mu} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi}_{\mu}}\left[\mathbb{E}_{a\sim\pi(\cdot|s)}A^{\pi'}(s,a)\right]$$

Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$

State visitation: $d^{\pi}_{\mu}(s) = (1-\gamma)\sum_{h=0}^{\infty}\gamma^h \mathbb{P}^{\pi}_h(s; s_0)$

Performance Difference Lemma:

What's the perf difference between π & π'?

$$V^{\pi}_{\mu} - V^{\pi'}_{\mu} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi}_{\mu}}\left[\mathbb{E}_{a\sim\pi(\cdot|s)}A^{\pi'}(s,a)\right]$$

The adv against π' averaged over the state distribution of π

Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$ What if r is unknown

State visitation: $d_\mu^\pi(s) = (1-\gamma)\sum_{h=0}^{\infty}\gamma^h \mathbb{P}_h^\pi(s; s_0)$

Performance Difference Lemma:

What's the perf difference between π & π'?

$$V_\mu^\pi - V_\mu^{\pi'} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^\pi}\left[\mathbb{E}_{a\sim\pi(\cdot|s)}A^{\pi'}(s,a)\right]$$

The adv against π' averaged over the state distribution of π

Outline for today:

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC

An Autonomous Land Vehicle In A Neural Network *[Pomerleau, NIPS '88]*

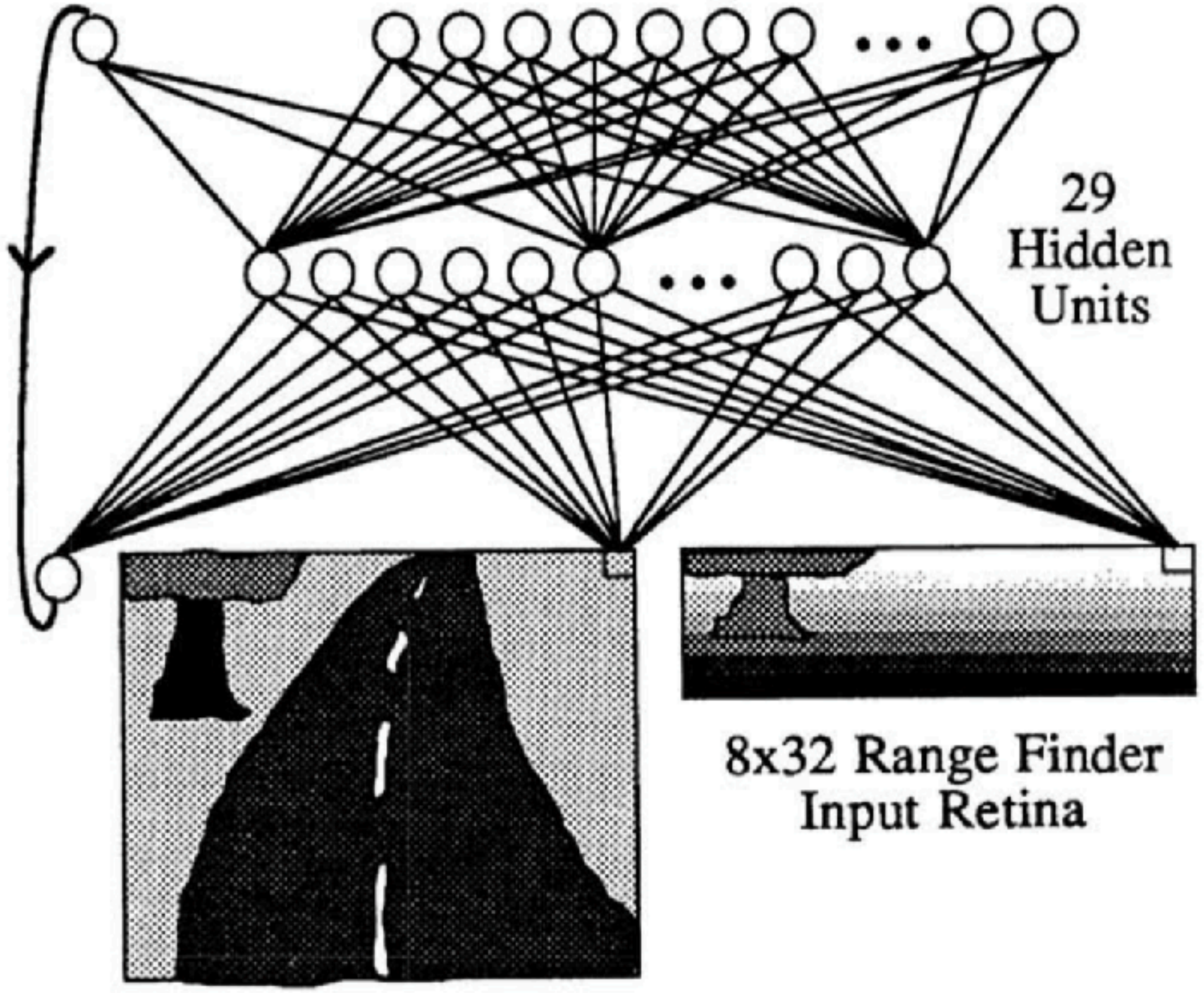
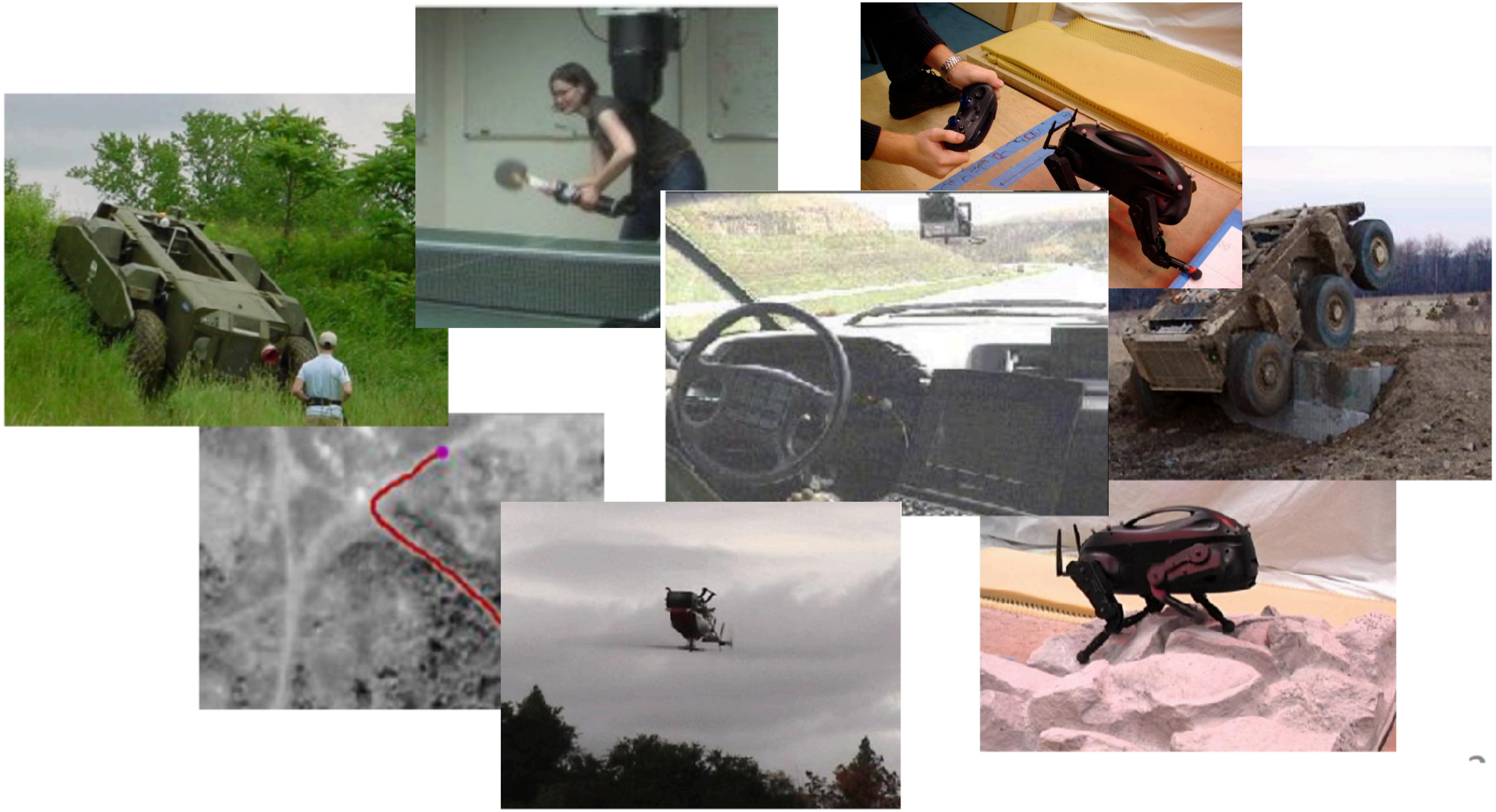


Figure 1: ALVINN Architecture

Imitation Learning



Imitation Learning

Imitation Learning

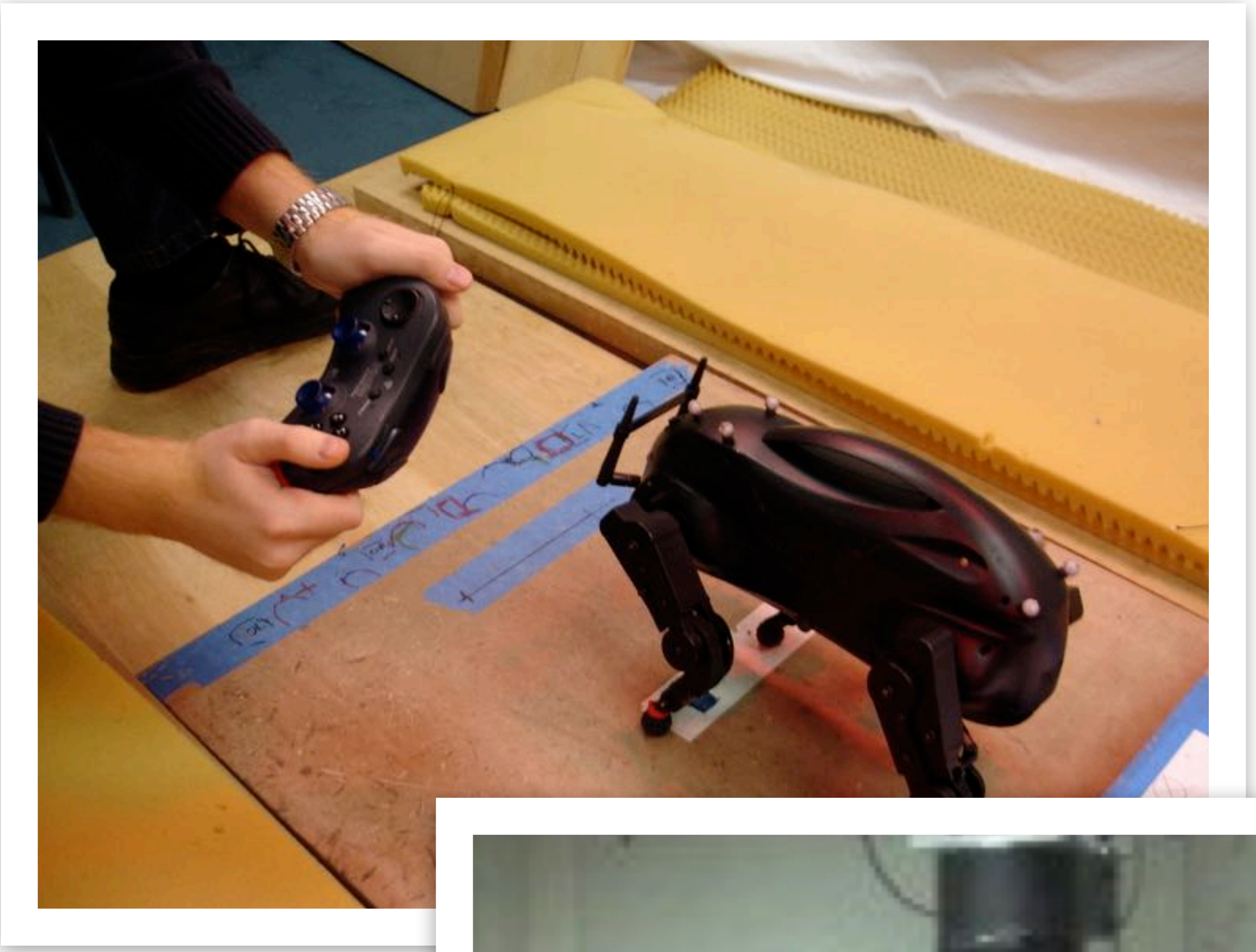
Imitation Learning

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 actions

Learning to Drive by Imitation

[Pomerleau89, Saxena05, Ross11a]

Input:

Camera Image

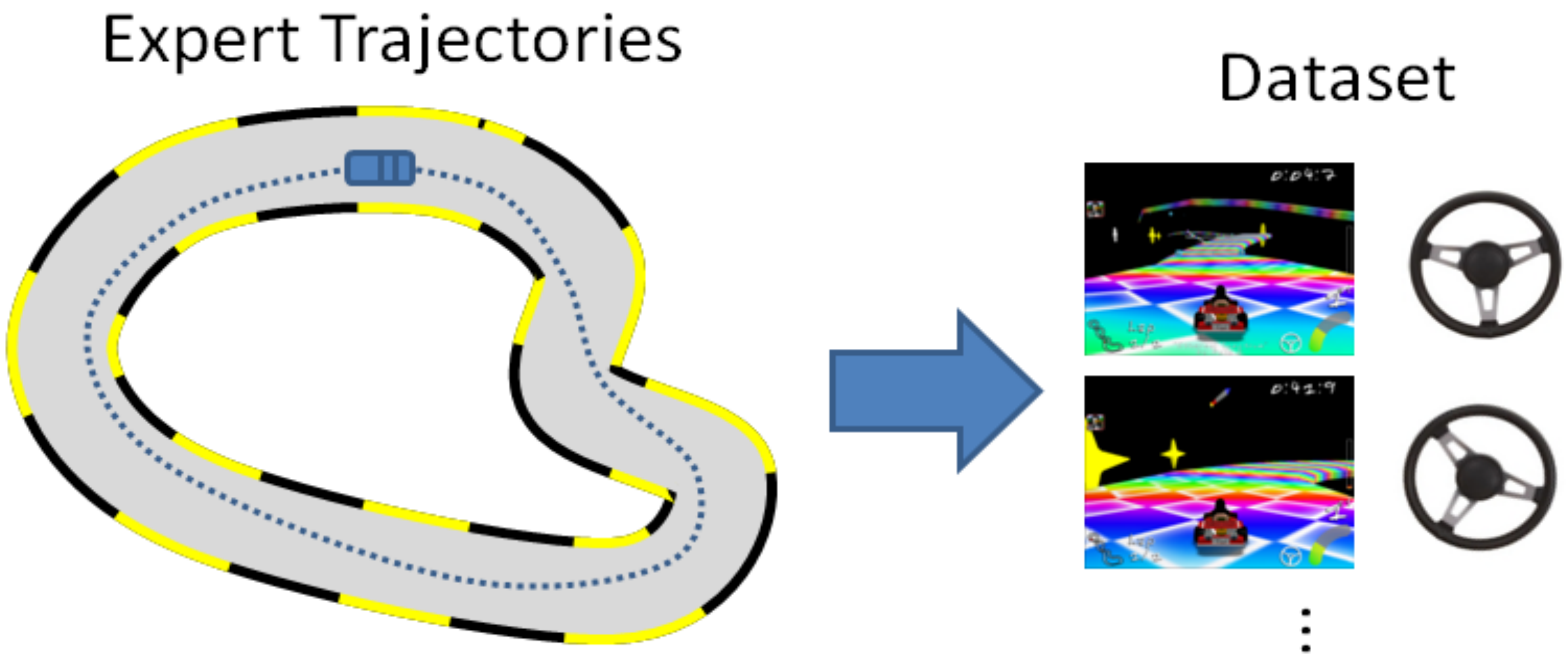
Policy

Output:

Steering Angle
in [-1, 1]

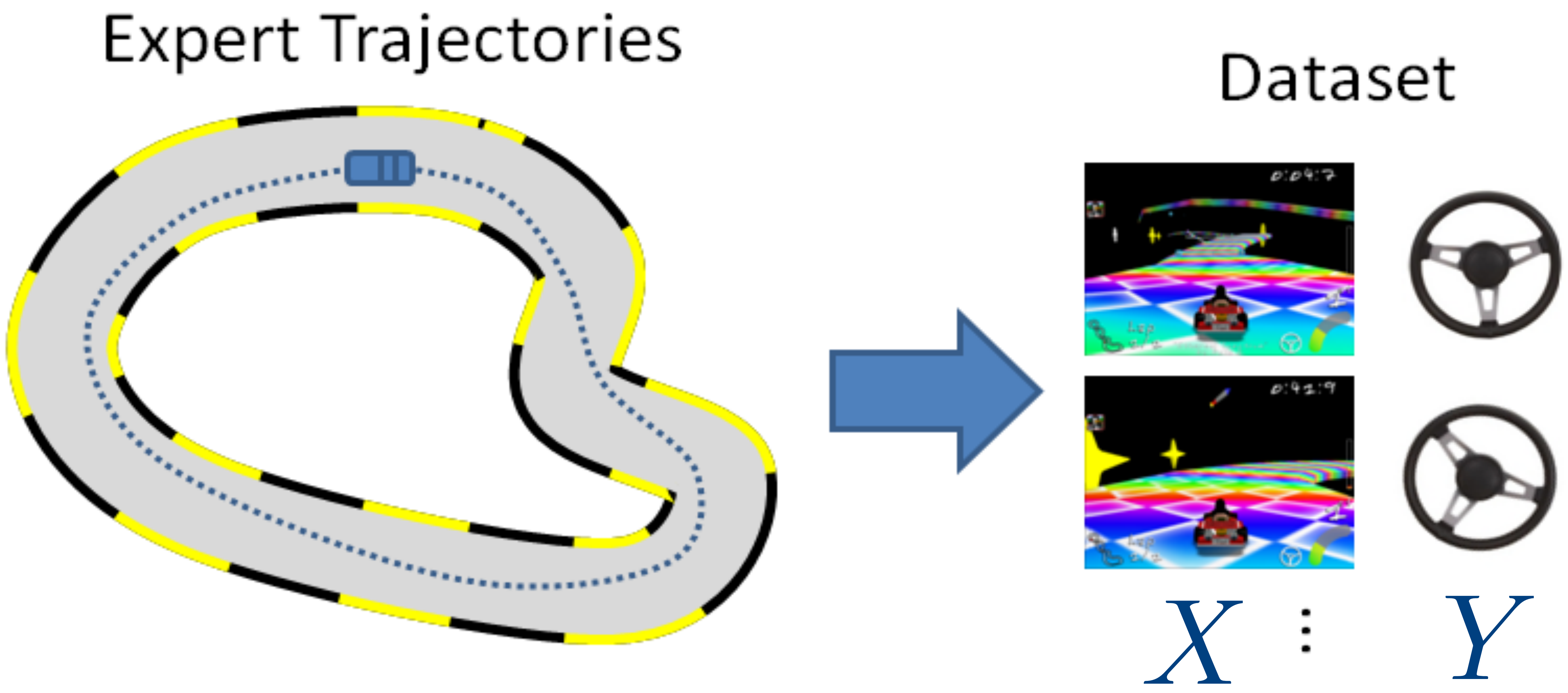
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



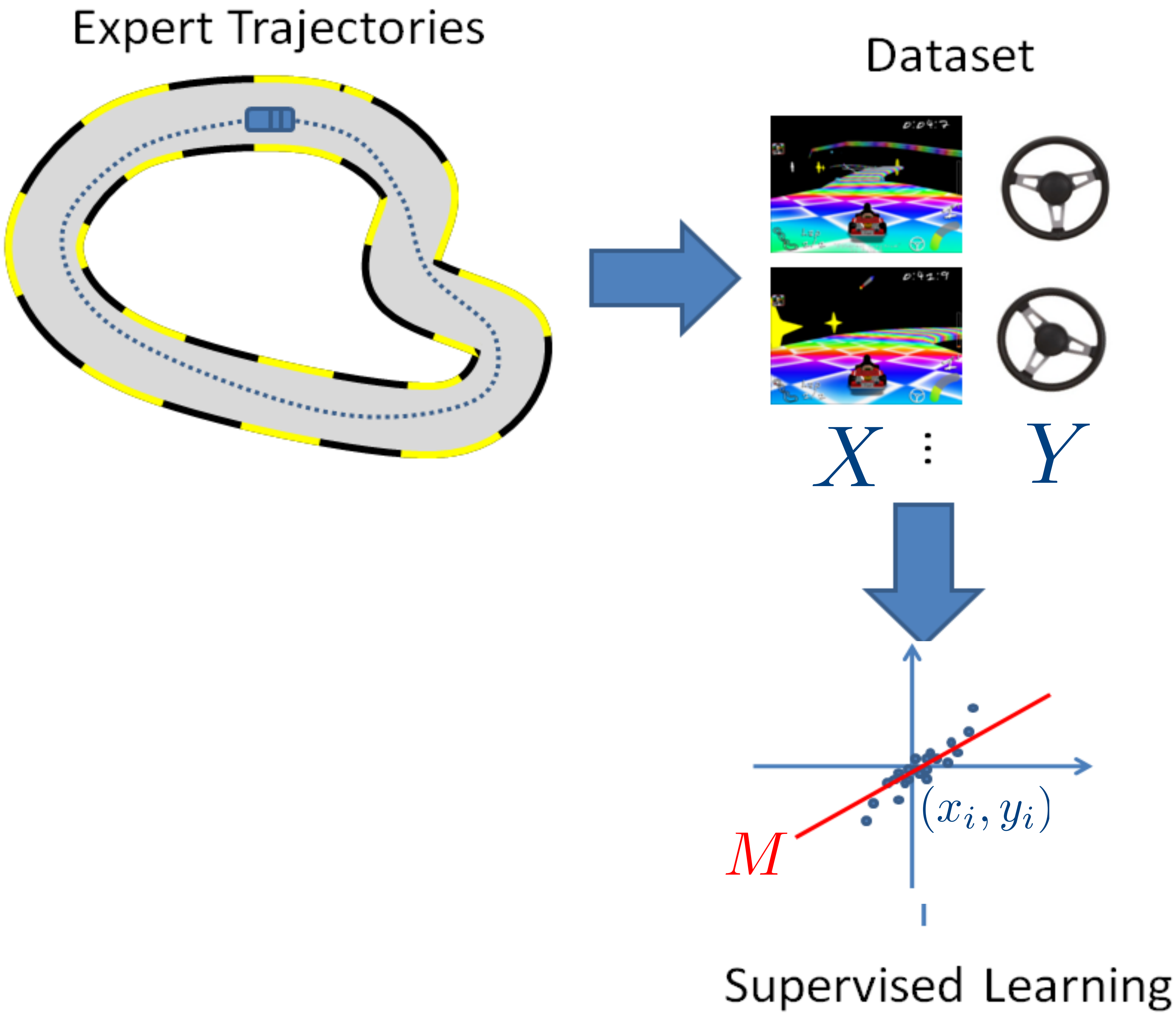
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



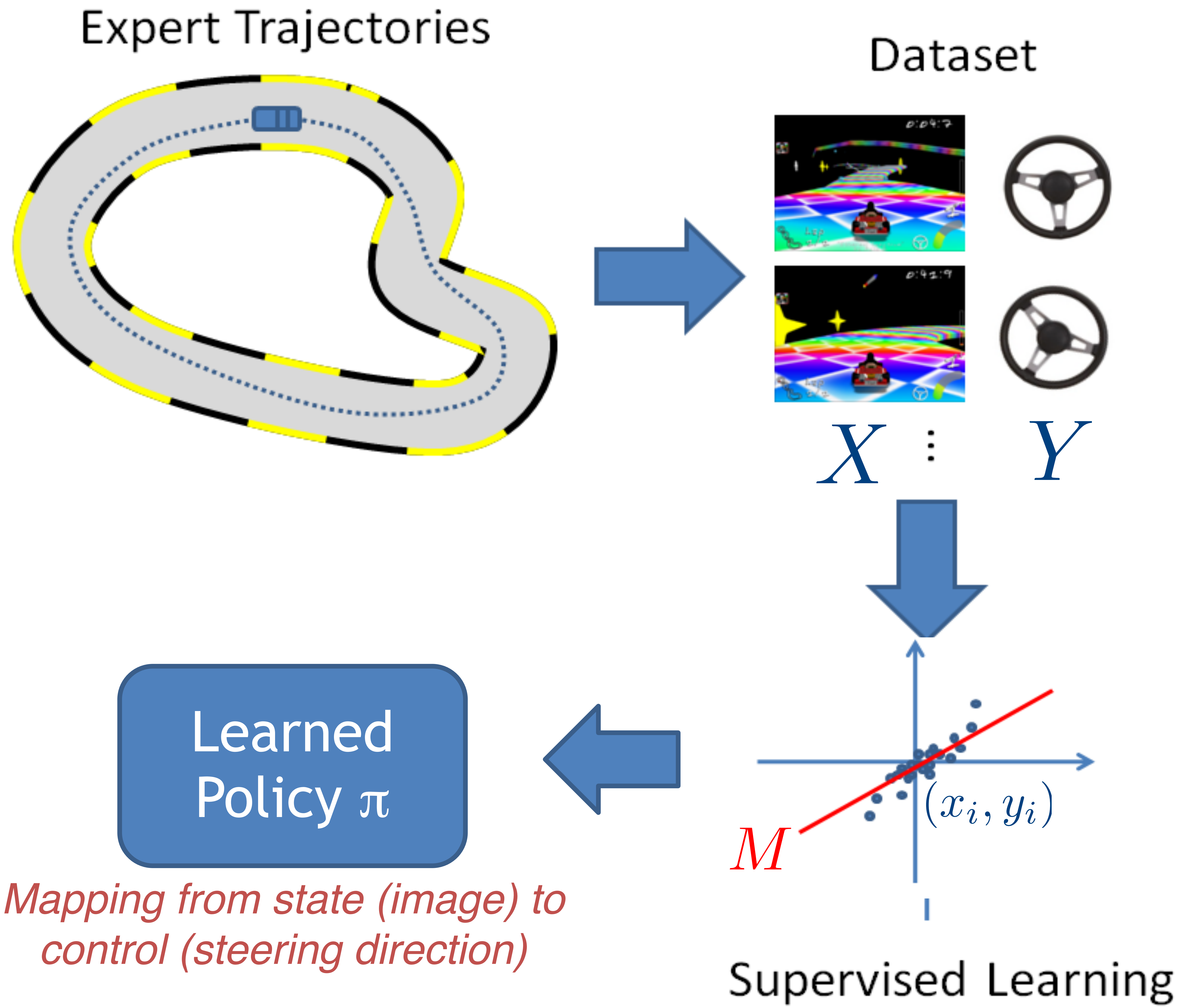
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning




Ready!


Ready!


Ready!

Outline

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC

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

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

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

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

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

For simplicity, let's assume expert is a (nearly) optimal policy $\pi^\star$

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

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

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

For simplicity, let's assume expert is a (nearly) optimal policy $\pi^\star$

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

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

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

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

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

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

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:

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:

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

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:

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

Many choices of loss functions:

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:

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

Many choices of loss functions:

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

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:

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

Many choices of loss functions:

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

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

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

Note that here training and testing mismatch at this stage!

Analysis

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

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

Analysis

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

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

$$(1-\gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s))$$

Analysis

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

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

$$(1-\gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s))$$

$$= \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s)) - \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \widehat{\pi}(s))$$

Analysis

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

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

$$(1-\gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s))$$

$$= \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s)) - \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \widehat{\pi}(s))$$

$$\leq \mathbb{E}_{s\sim d^{\pi^\star}} \frac{2}{1-\gamma} \mathbf{1}\left\{\widehat{\pi}(s) \neq \pi^\star(s)\right\}$$

Analysis

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

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

$$(1-\gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s))$$

$$= \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s)) - \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \widehat{\pi}(s))$$

$$\leq \mathbb{E}_{s\sim d^{\pi^\star}} \frac{2}{1-\gamma} \mathbf{1}\left\{\widehat{\pi}(s) \neq \pi^\star(s)\right\}$$

$$\leq \frac{2}{1-\gamma}\epsilon$$

Analysis

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

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

The quadratic amplification is annoying

$$(1-\gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s))$$

$$= \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \pi^\star(s)) - \mathbb{E}_{s\sim d^{\pi^\star}} A^{\widehat{\pi}}(s, \widehat{\pi}(s))$$

$$\leq \mathbb{E}_{s\sim d^{\pi^\star}} \frac{2}{1-\gamma} \mathbf{1}\left\{\widehat{\pi}(s) \neq \pi^\star(s)\right\}$$

$$\leq \frac{2}{1-\gamma}\epsilon$$

Outline

1. Introduction of Imitation Learning

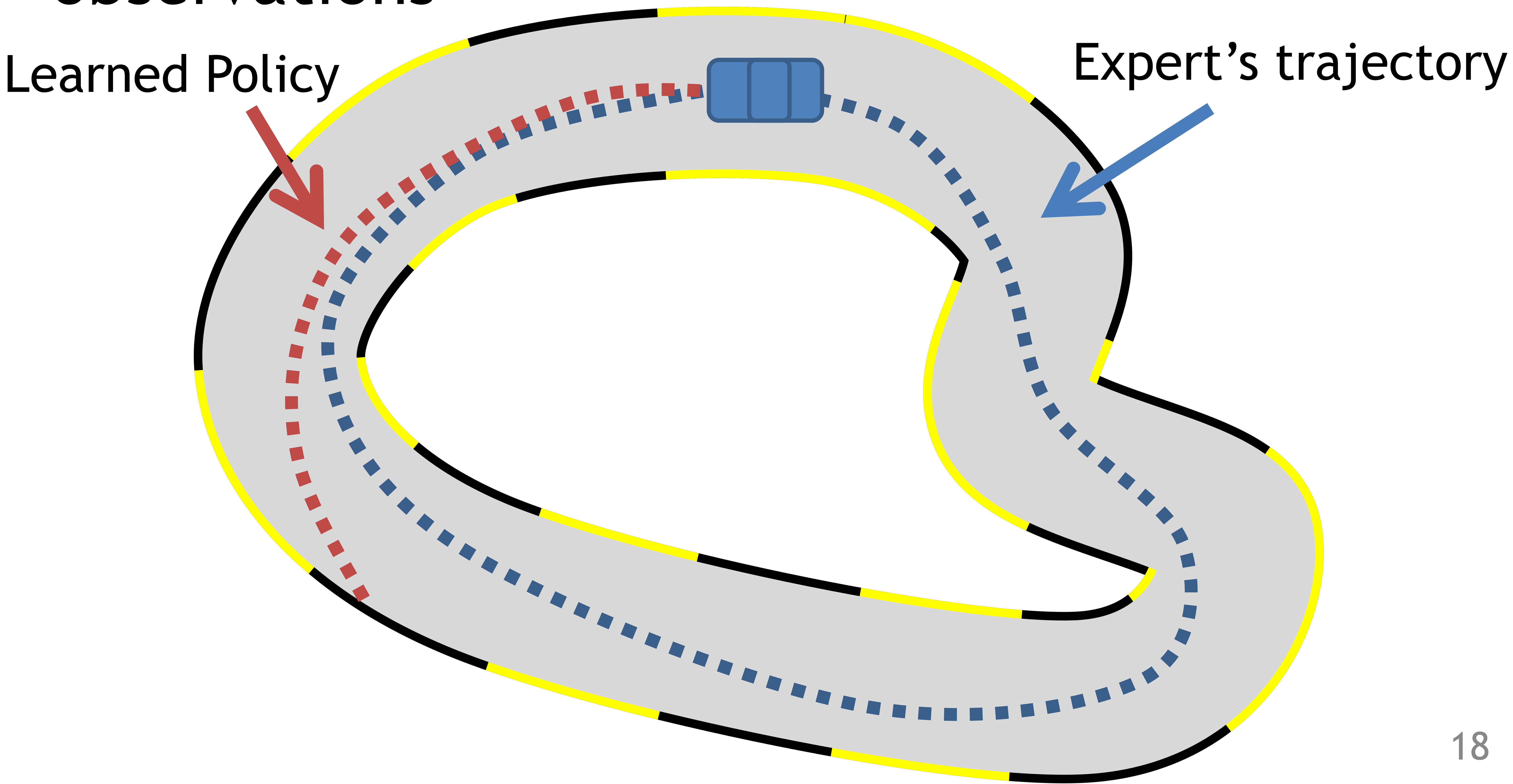
2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC

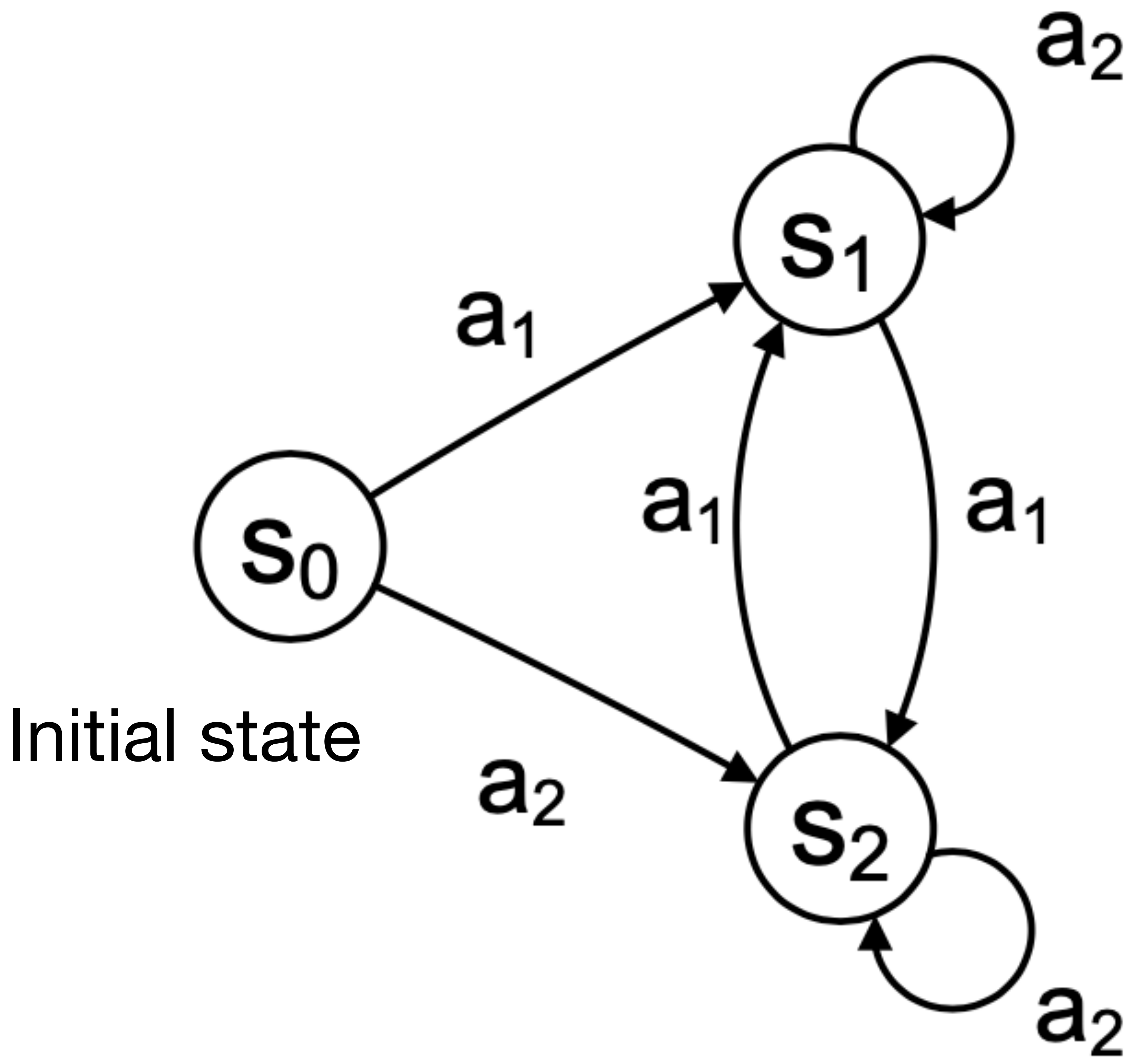
What could go wrong?

[Pomerleau89,Daume09]

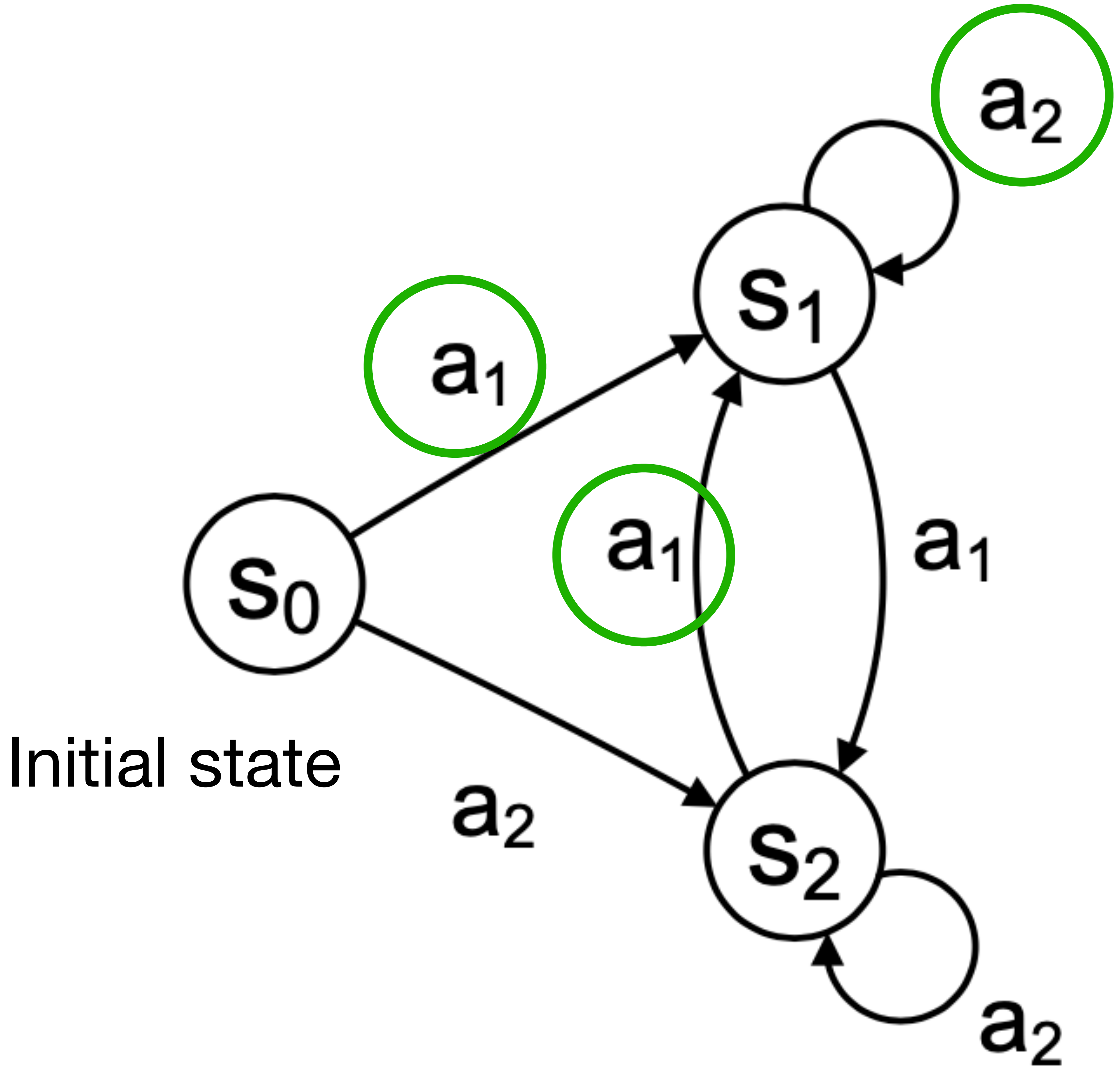
- Predictions affect future inputs/ observations



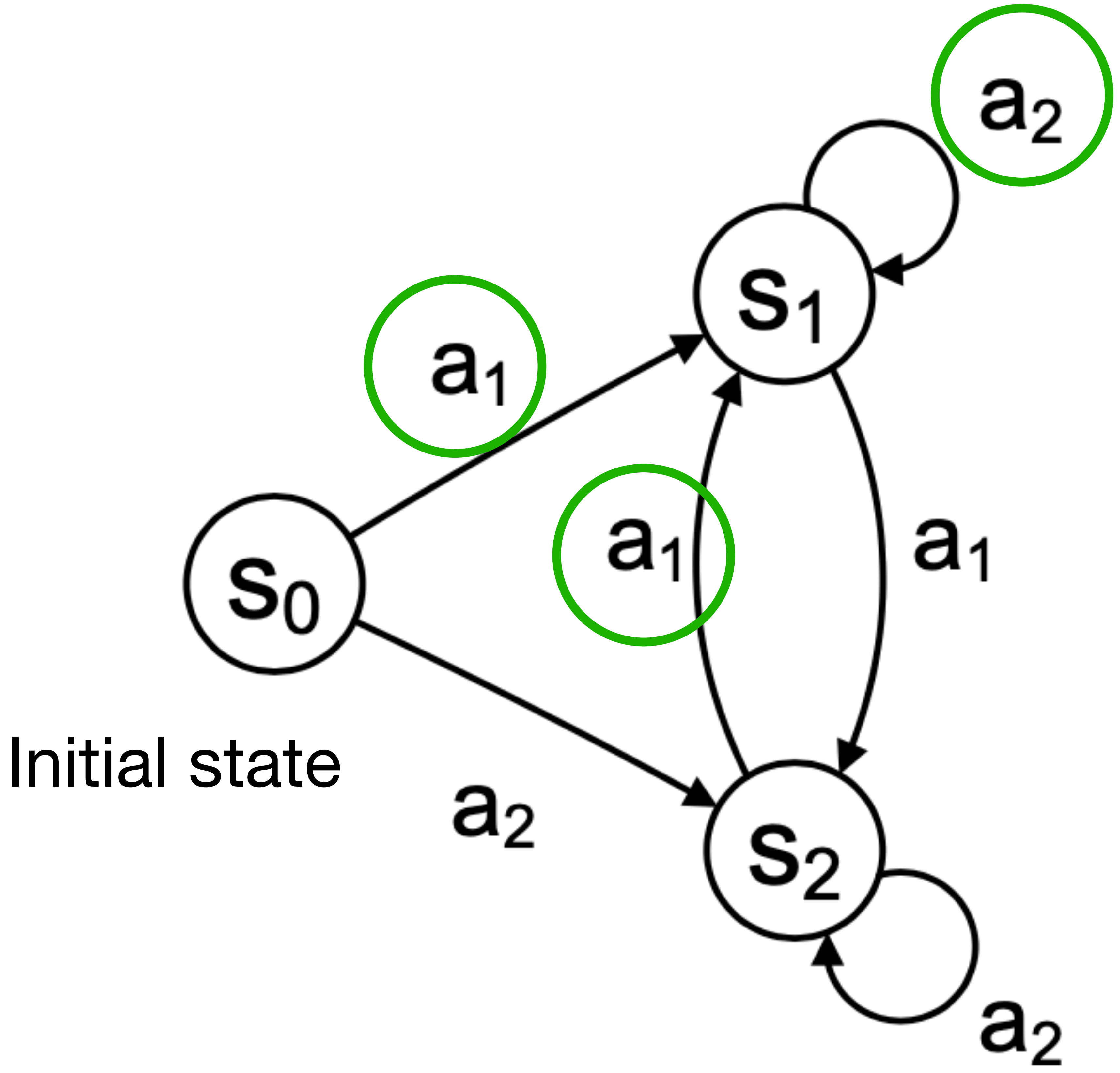
Distribution Shift: Example



Distribution Shift: Example



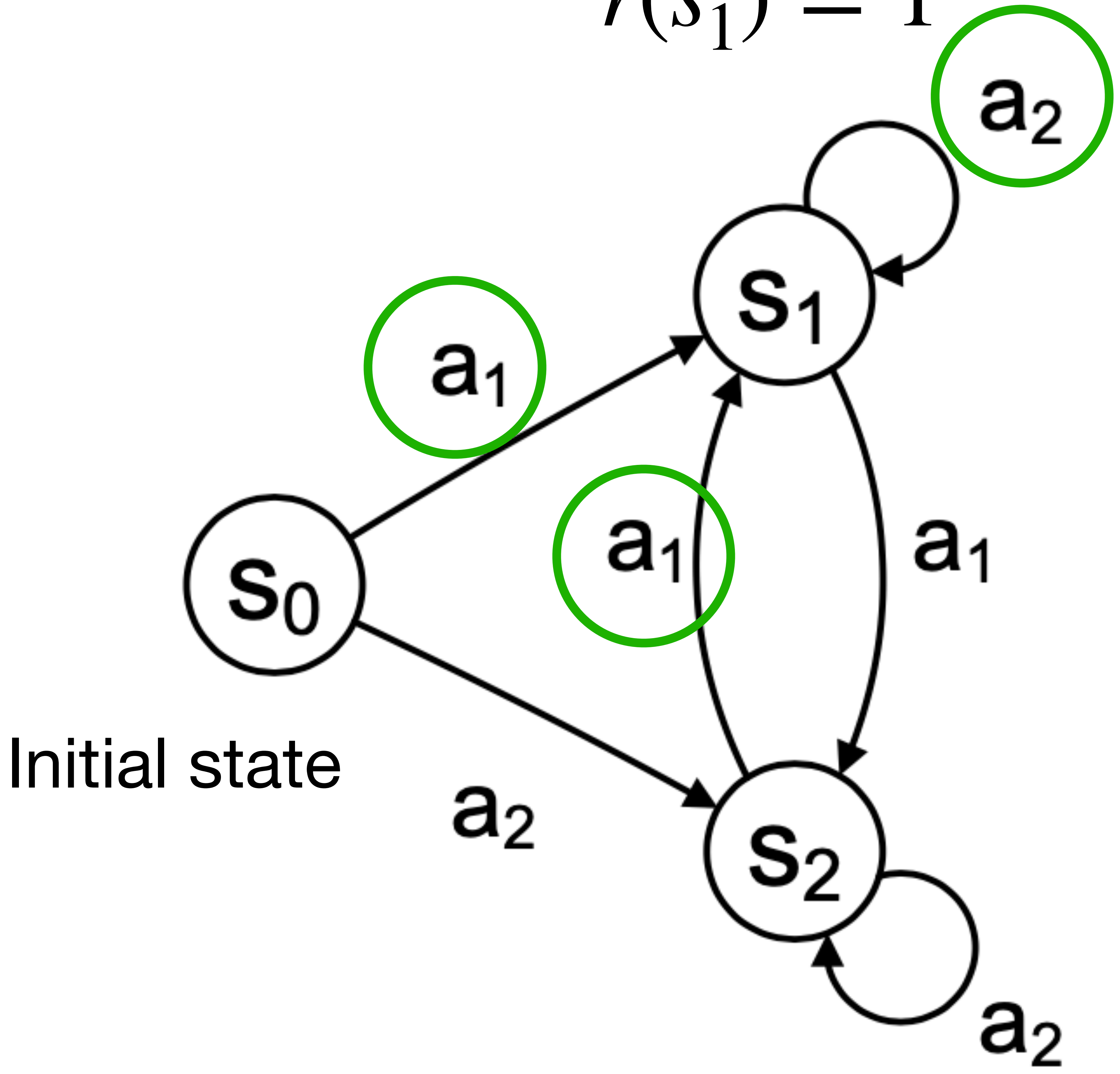
Distribution Shift: Example



$$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$$

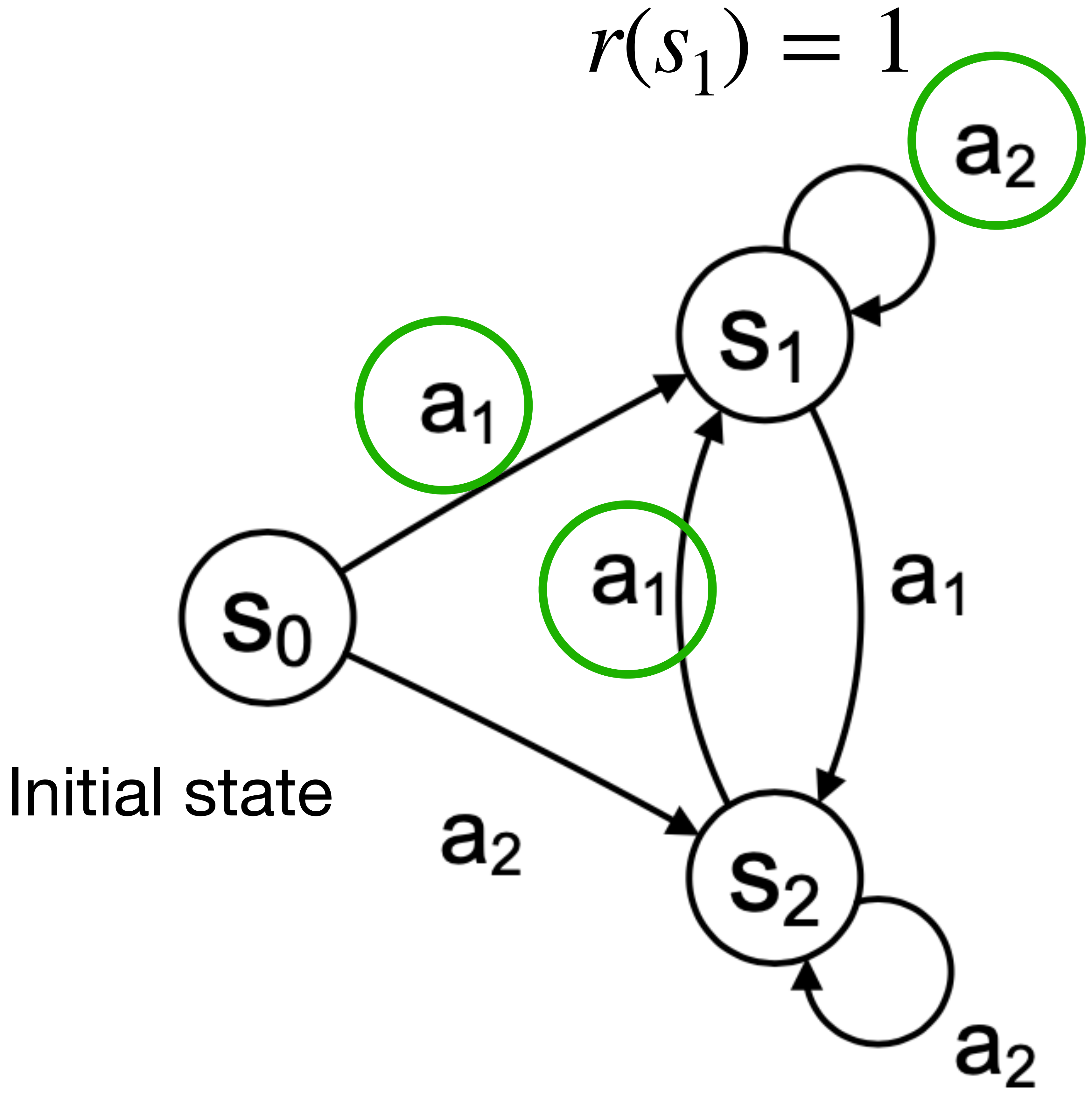
Distribution Shift: Example

$r(s_1) = 1$



$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$

Distribution Shift: Example

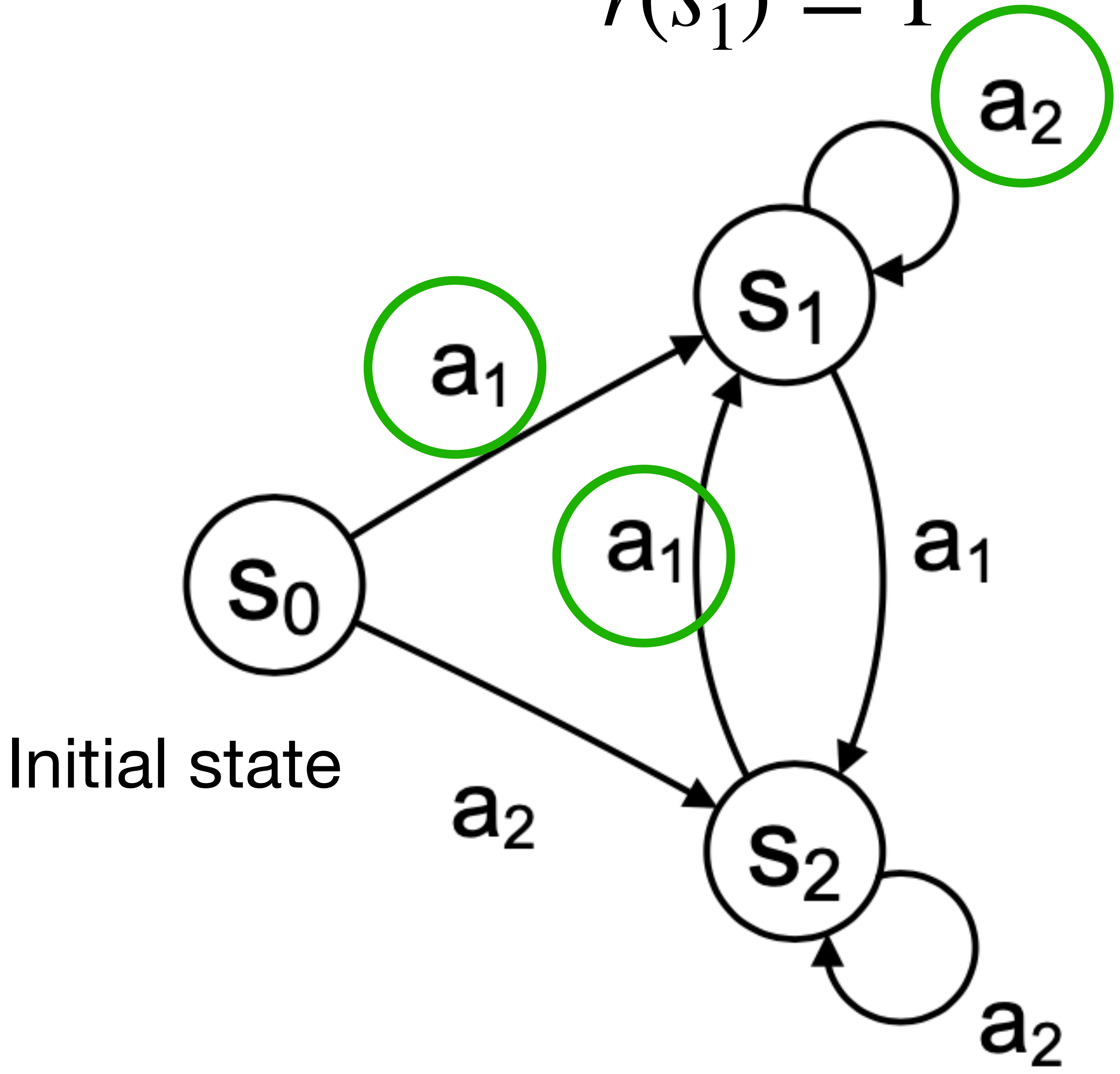


$$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$$

$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

Distribution Shift: Example

$r(s_1) = 1$



$$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$$

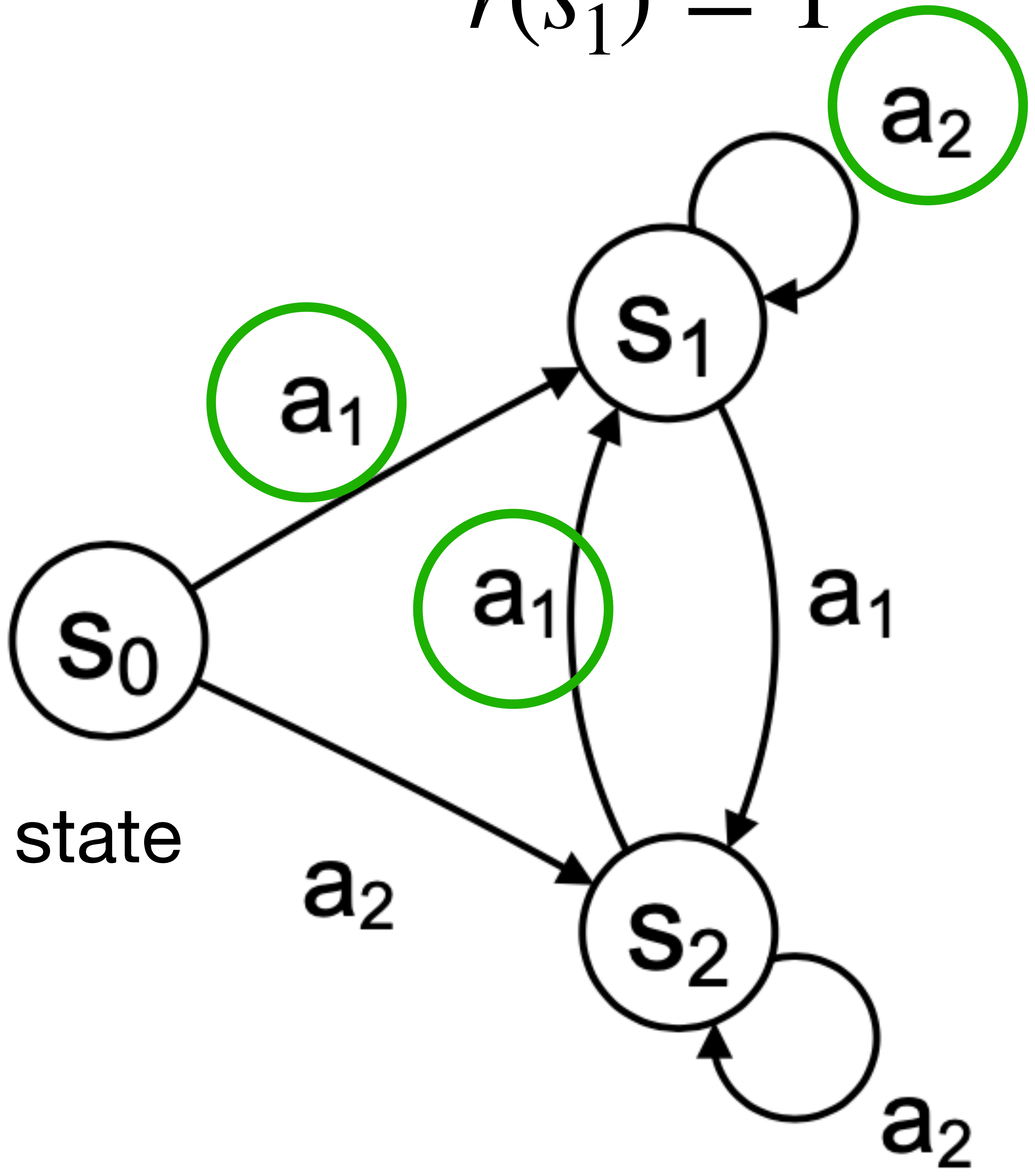
$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

Assume SL returned such policy $\widehat{\pi}$

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

Distribution Shift: Example

$r(s_1) = 1$



Initial state

$d^{\pi^\star}_{s_0}(s_0) = 1 - \gamma,\ d^{\pi^\star}_{s_0}(s_1) = \gamma,\ d^{\pi^\star}_{s_0}(s_2) = 0$

$V^{\pi^\star}_{s_0} = \dfrac{\gamma}{1-\gamma}$

Assume SL returned such policy $\widehat{\pi}$

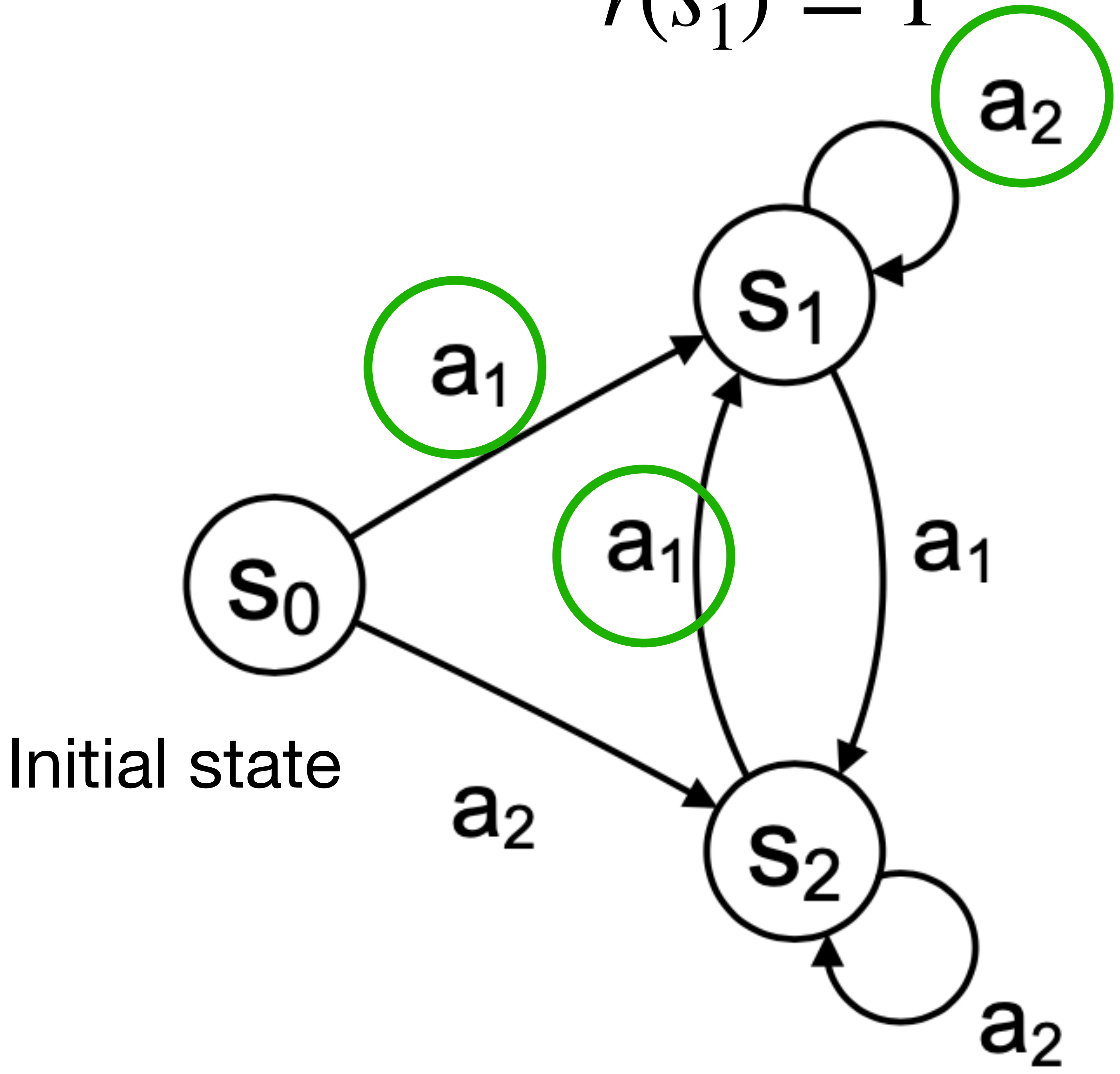
$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - \epsilon/(1-\gamma) \\ a_2 & \text{w/ prob } \epsilon/(1-\gamma) \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

We will have good supervised learning error:

$$\mathbb{E}_{s \sim d^{\pi^\star}_{s_0}} \mathbb{E}_{a \sim \widehat{\pi}(\cdot|s)} \mathbf{1}\left(a \neq \pi^\star(s)\right) = \epsilon$$

Distribution Shift: Example

$r(s_1) = 1$



$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$

$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

Assume SL returned such policy $\widehat{\pi}$

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

We will have good supervised learning error:

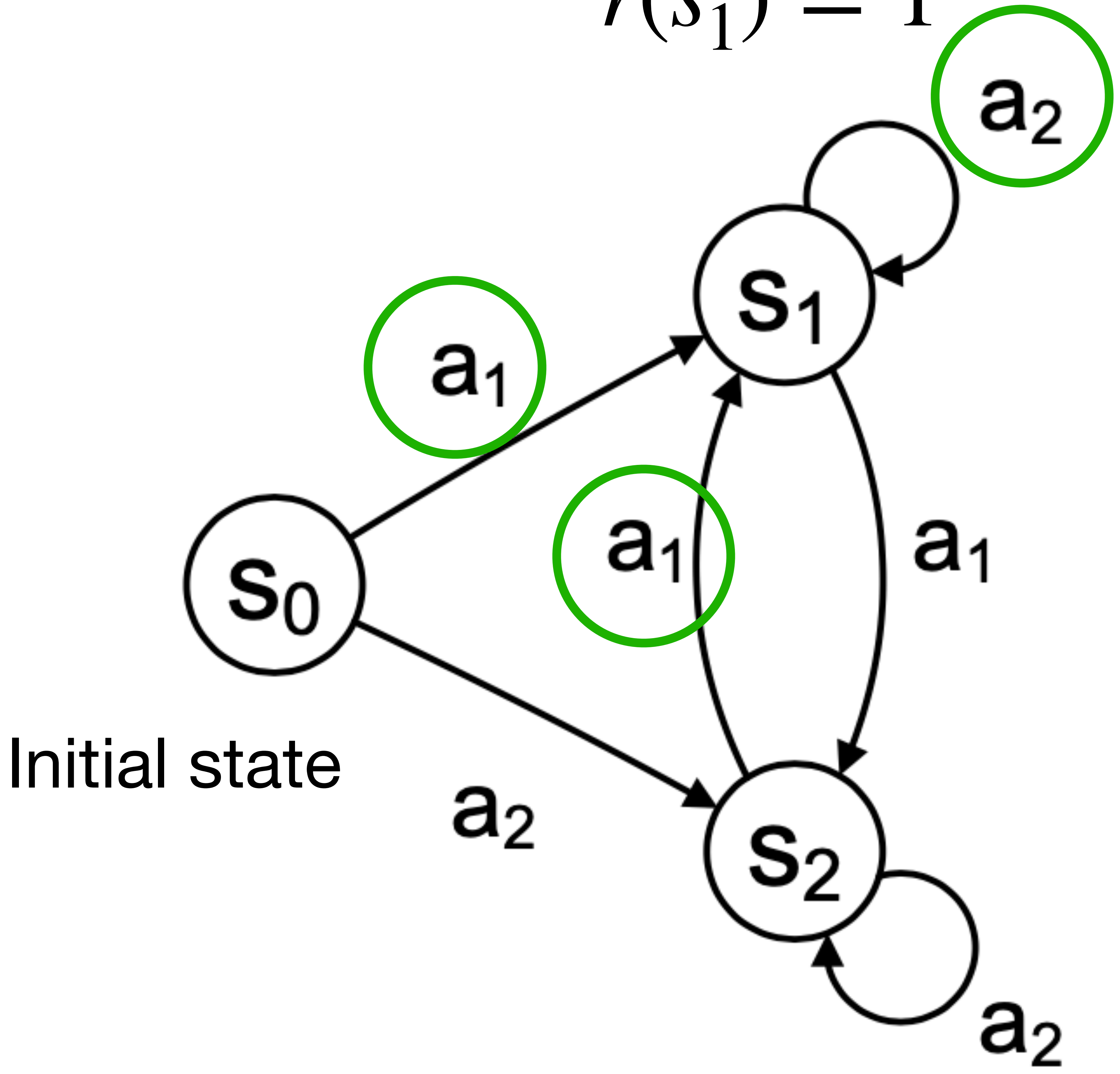
$$\mathbb{E}_{s \sim d_{s_0}^{\pi^\star}} \mathbb{E}_{a \sim \widehat{\pi}(\cdot|s)} \mathbf{1}\left(a \neq \pi^\star(s)\right) = \epsilon$$

But we have quadratic error in performance:

$$V_{s_0}^{\widehat{\pi}} = \frac{\gamma}{1-\gamma} - \frac{\epsilon\gamma}{(1-\gamma)^2} = V_{s_0}^{\pi^\star} - \frac{\epsilon\gamma}{(1-\gamma)^2}$$

Distribution Shift: Example

$r(s_1) = 1$



$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$

$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

Assume SL returned such policy $\widehat{\pi}$

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

We will have good supervised learning error:

$$\mathbb{E}_{s \sim d_{s_0}^{\pi^\star}} \mathbb{E}_{a \sim \widehat{\pi}(\cdot|s)} \mathbf{1}\left(a \neq \pi^\star(s)\right) = \epsilon$$

But we have quadratic error in performance:

$$V_{s_0}^{\widehat{\pi}} = \frac{\gamma}{1-\gamma} - \frac{\epsilon\gamma}{(1-\gamma)^2} = V_{s_0}^{\pi^\star} - \frac{\epsilon\gamma}{(1-\gamma)^2}$$

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

An Autonomous Land Vehicle In A Neural Network *[Pomerleau, NIPS ‘88]*

“If the network is not presented with sufficient variability in its training exemplars to cover the conditions it is likely to encounter…[it] will perform poorly”

An Autonomous Land Vehicle In A Neural Network
[Pomerleau, NIPS ‘88]

“If the network is not presented with sufficient variability in its training exemplars to cover the conditions it is likely to encounter…[it] will perform poorly”

Summary for Today:

1. The most common imitation learning algorithm: BC

A reduction to supervised Learning, e.g., training classifier from $s^\star \sim d_\mu^{\pi^\star}, a^\star = \pi^\star(s^\star)$

Summary for Today:

1. The most common imitation learning algorithm: BC

A reduction to supervised Learning, e.g., training classifier from $s^\star \sim d_\mu^{\pi^\star}, a^\star = \pi^\star(s^\star)$

2. Distribution shift:

When execute the learned policy,
we may deviate from the expert trajectories,
causing compounding error

Summary for Today:

1. The most common imitation learning algorithm: BC

A reduction to supervised Learning, e.g., training classifier from $s^\star \sim d_\mu^{\pi^\star}, a^\star = \pi^\star(s^\star)$

2. Distribution shift:

When execute the learned policy,
we may deviate from the expert trajectories,
causing compounding error

3. **Again this demonstrates why RL/IL is harder than SL:**
we need to test our model on new data generated by our model