Interactive Imitation Learning

Announcement

This Thursday:

lecture will start at 9:50am
and office hour will end at 11:15am

Recap

Offline IL

Recap

Offline IL

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

Recap

Offline IL

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

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

Recap

The Behavior Cloning algorithm:

Choose regression (for continuous action) or classification loss $\ell(\pi(s), a)$, and perform SL:

$$\widehat{\pi} = \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi^\star(s_i^\star), a_i^\star)$$

Recap

The Behavior Cloning algorithm:

Choose regression (for continuous action) or classification loss $\ell(\pi(s), a)$, and perform SL:

$$\widehat{\pi} = \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi^\star(s_i^\star), a_i^\star)$$

Pros:

Simple, flexible, and often just works reasonably well

Recap

The Behavior Cloning algorithm:

Choose regression (for continuous action) or classification loss $\ell(\pi(s), a)$, and perform SL:

$$\widehat{\pi} = \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi^\star(s_i^\star), a_i^\star)$$

Pros:

Simple, flexible, and often just works reasonably well

Cons:

Distribution shift issue: $\widehat{\pi}$ does not know what to do outside expert's states

Question for today:

How to mitigate the distribution shift issue?

Solution:

Interactive Imitation Learning Setting

Solution:

Interactive Imitation Learning Setting

Key assumption:

we can query expert $\pi^\star$ at any time and any state during training

Solution:

Interactive Imitation Learning Setting

Key assumption:
we can query expert $\pi^\star$ at any time and any state during training

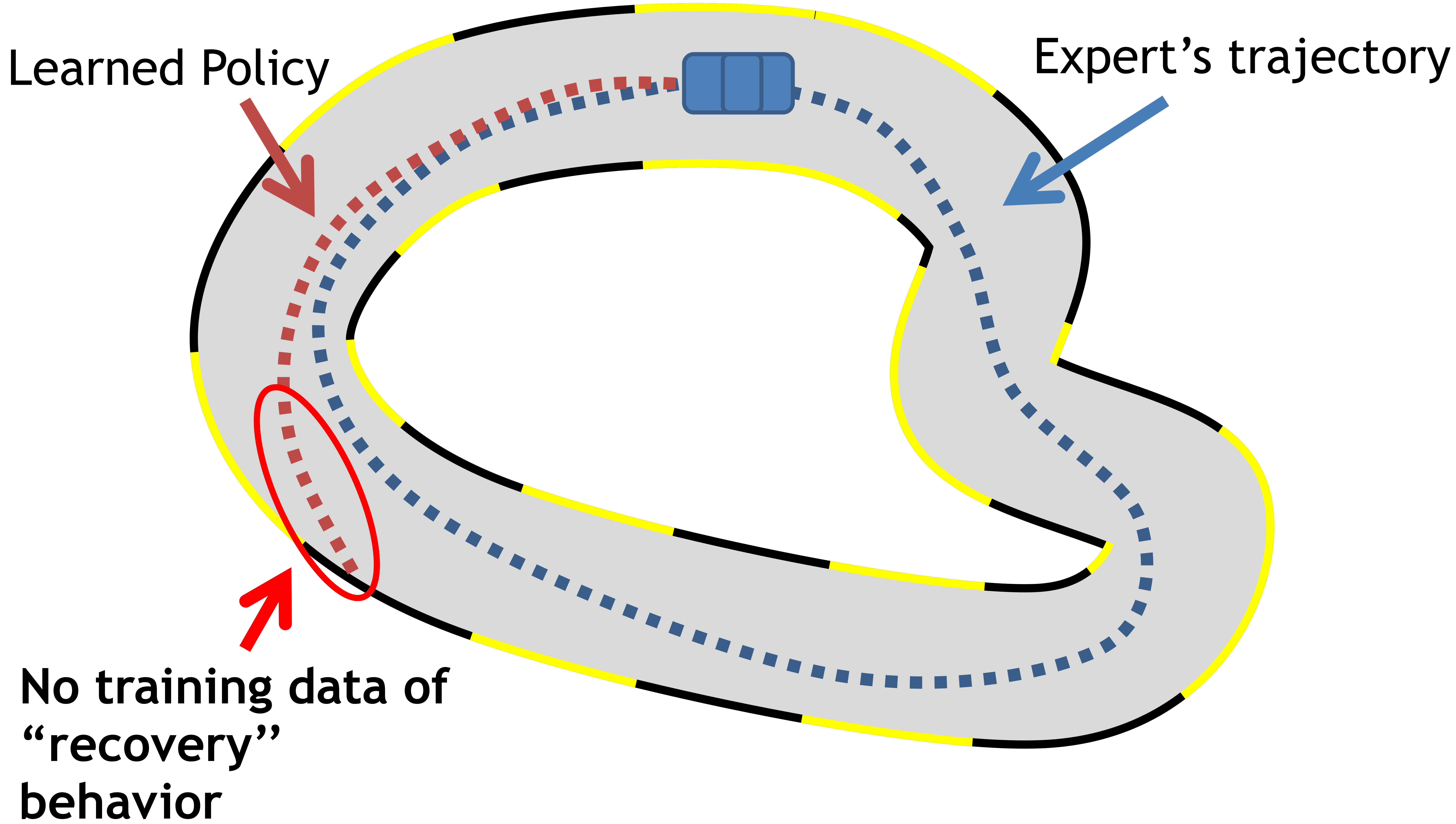
(Recall that previously we only had an offline dataset $\mathcal{D} = (s_i^\star, a_i^\star)_{i=1}^M \sim d_\mu^{\pi^\star}$)

Outline for today:

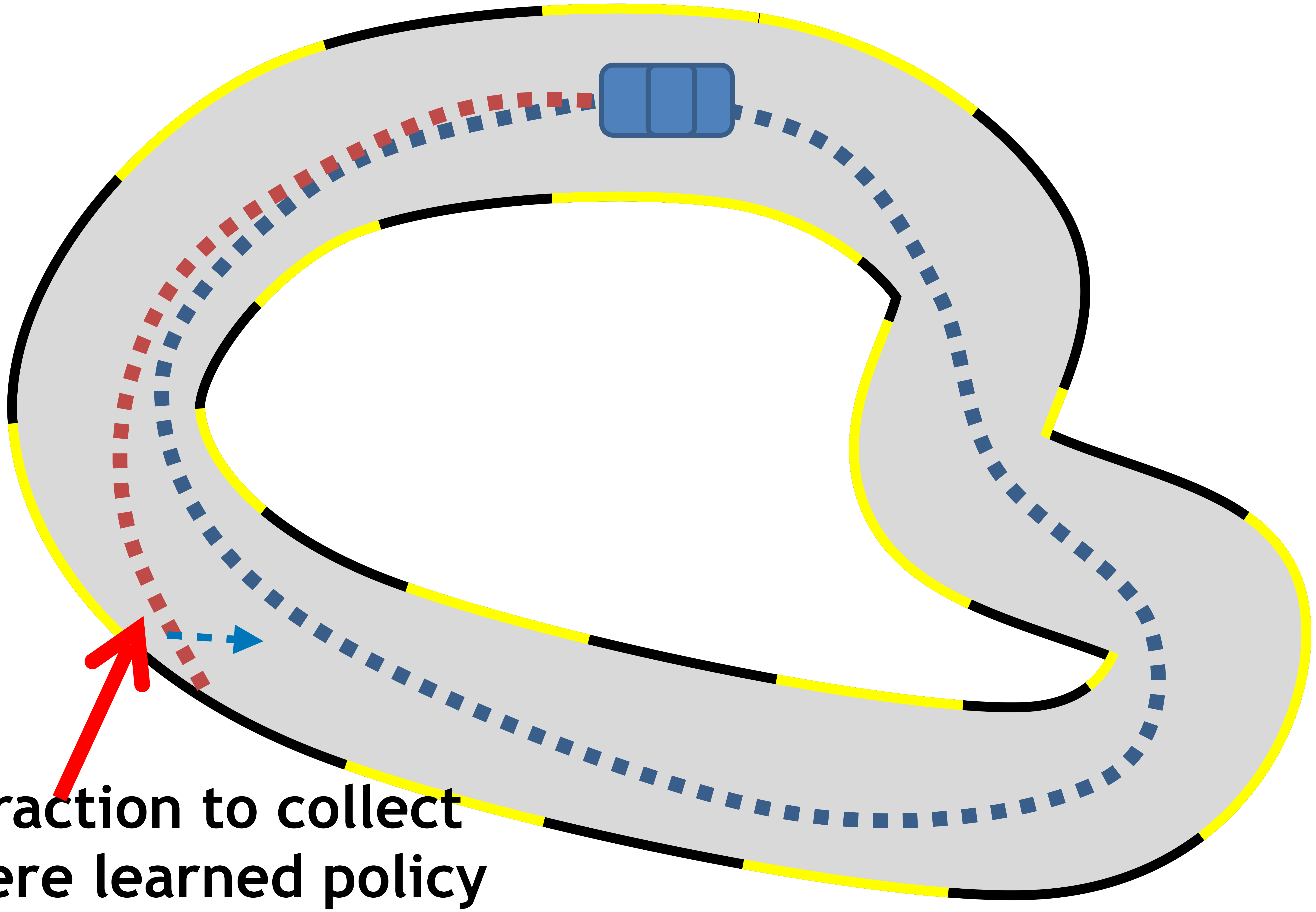
1. The DAgger (Data Aggregation) Algorithm

2. Analysis of DAgger: a quick intro to Online Learning

Recall the Main Problem from Behavior Cloning:

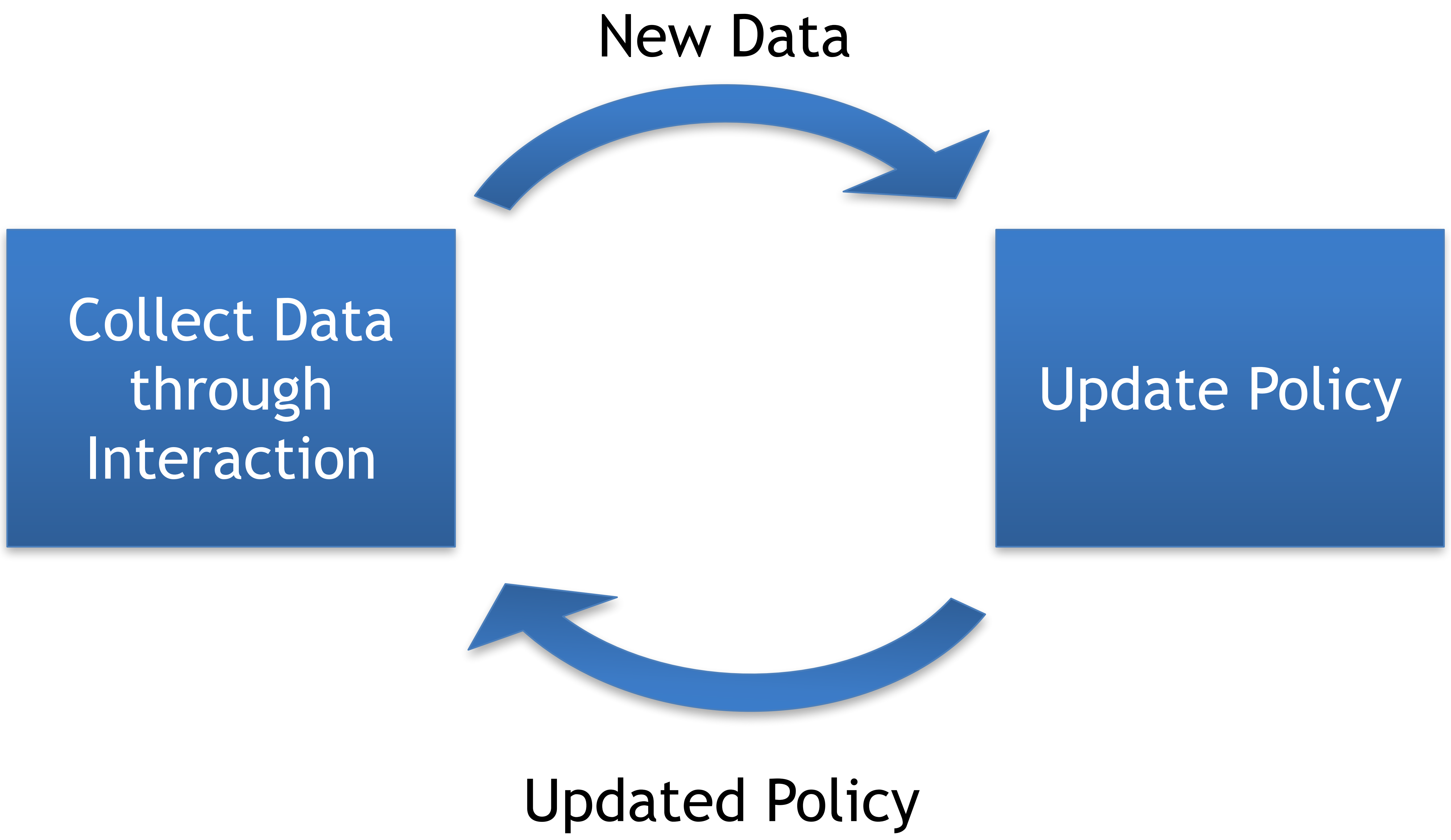


Intuitive solution: **Interaction**



Use interaction to collect data where learned policy goes

General Idea: Iterative Interactive Approach

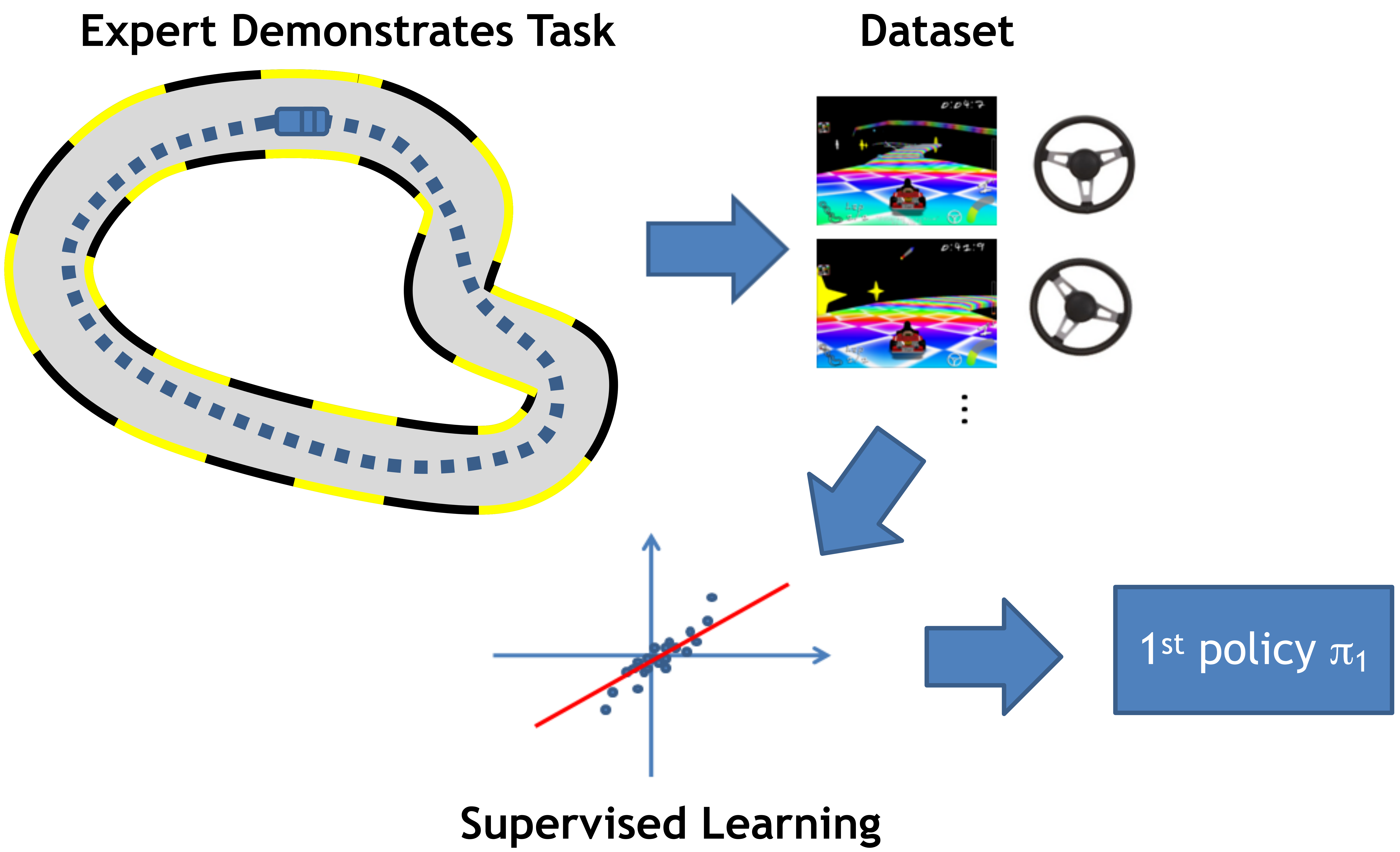


All DAgger slides credit: Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Dataset Aggregation

[Ross11a]

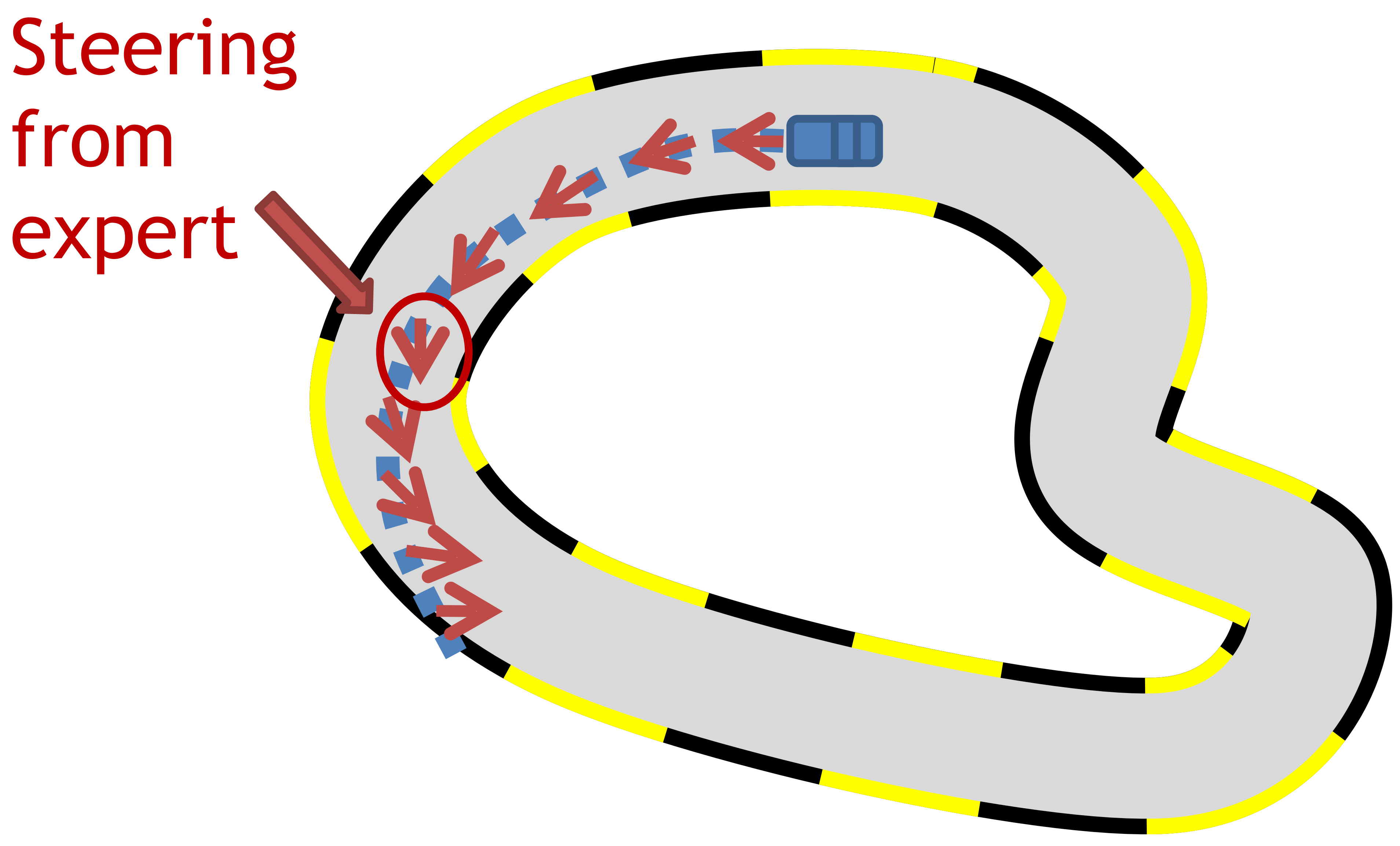
0th iteration



DAgger: Dataset Aggregation [Ross11a]

1st iteration

Execute π_1 and Query Expert

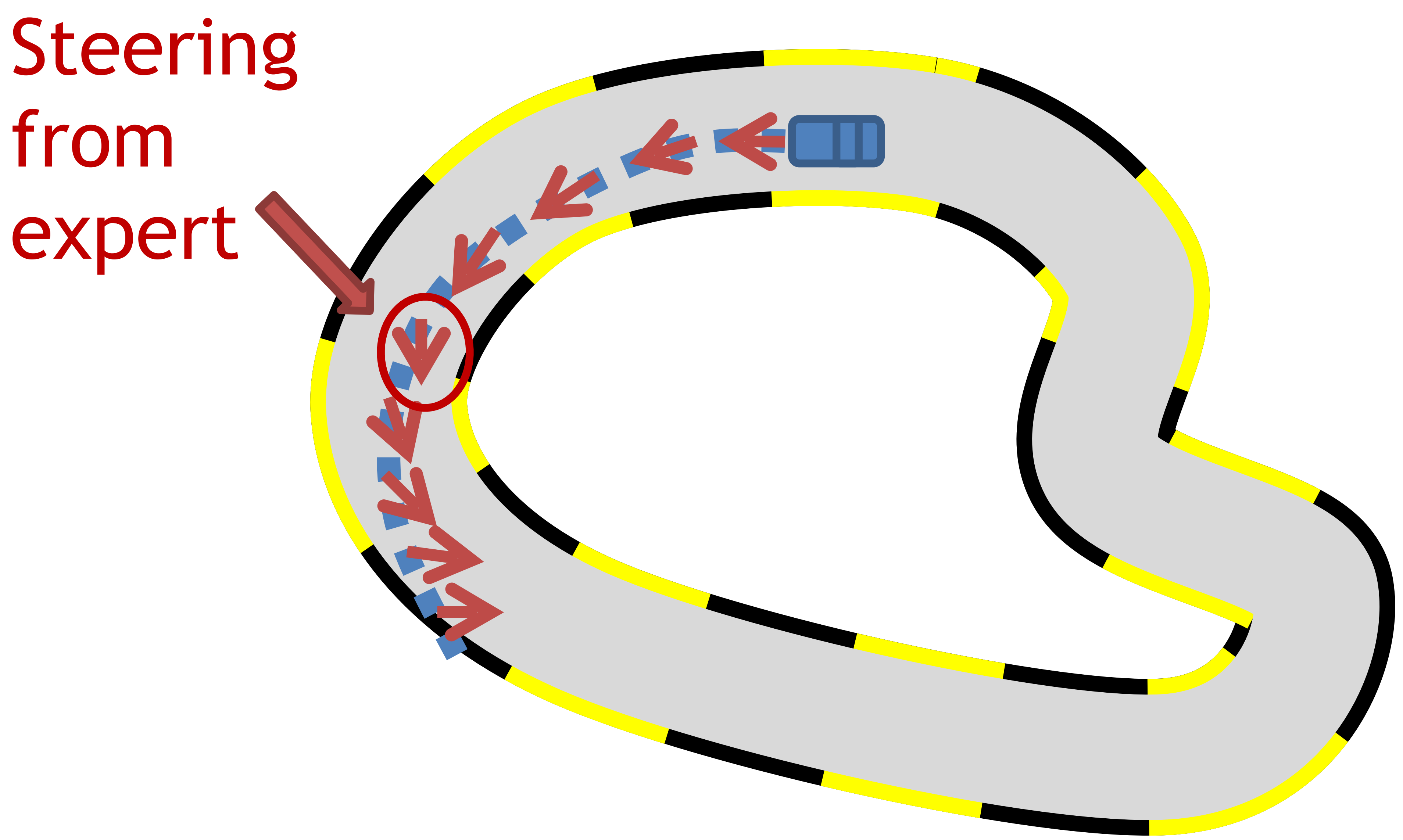


DAgger: Dataset Aggregation

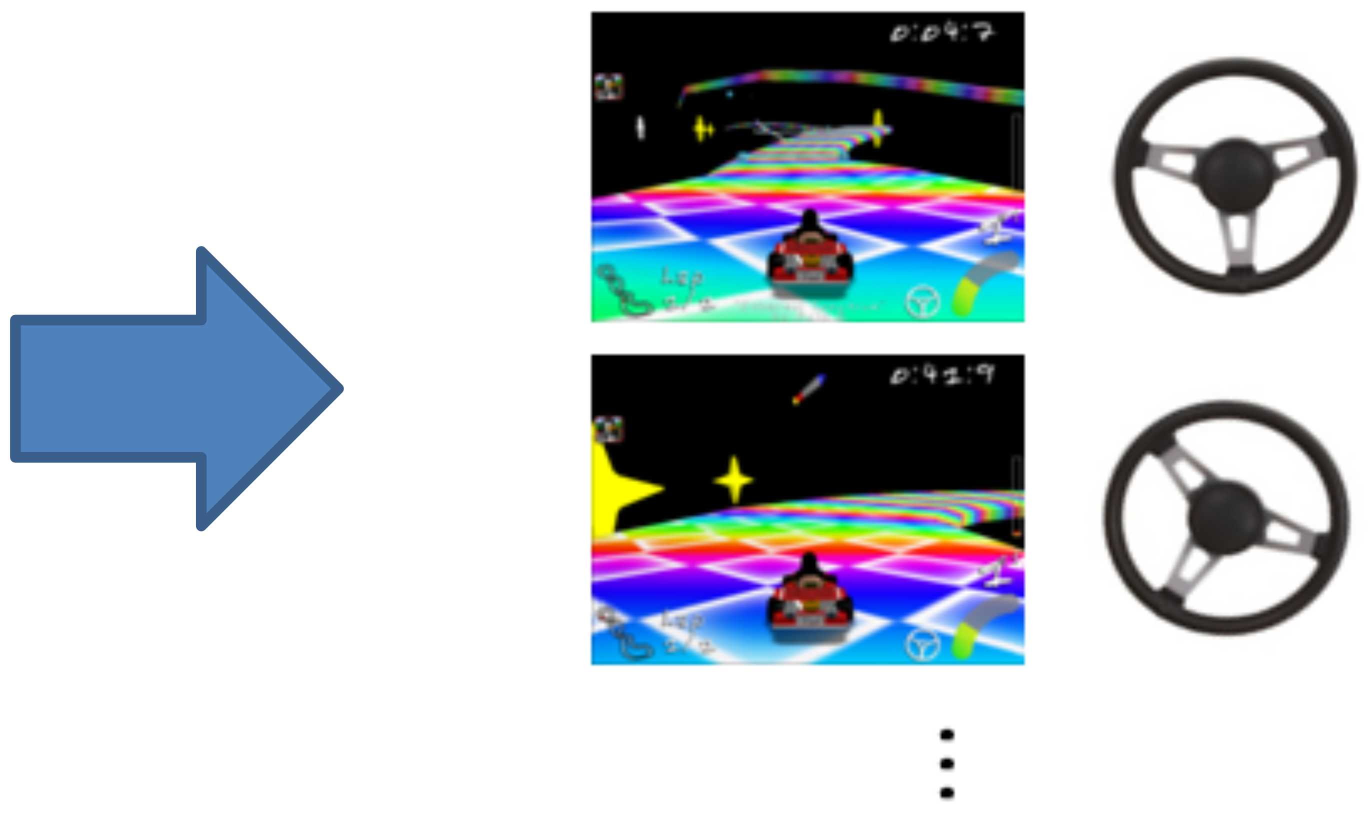
[Ross11a]

1st iteration

Execute π_1 and Query Expert



New Data

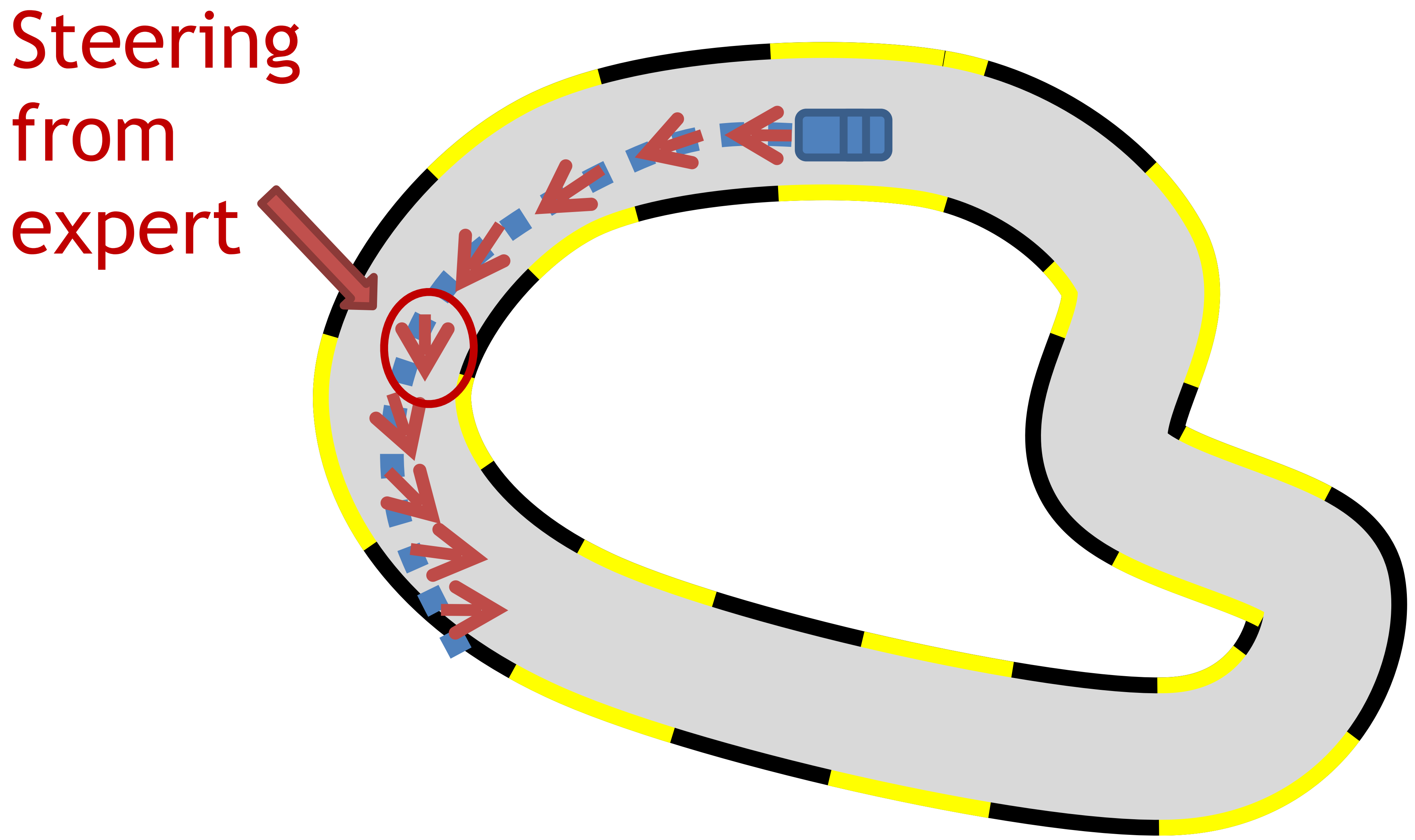


DAgger: Dataset Aggregation

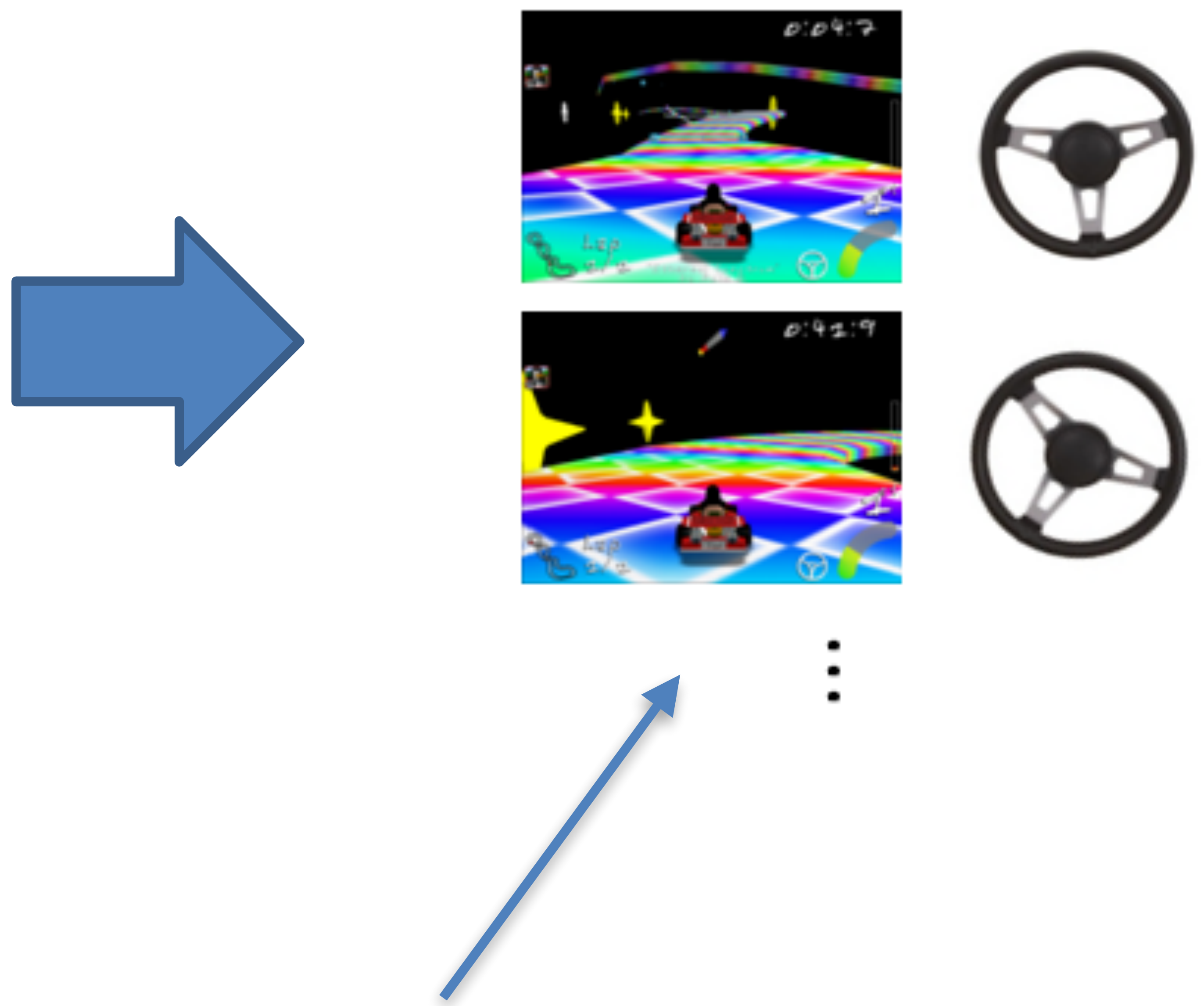
[Ross11a]

1st iteration

Execute π_1 and Query Expert



New Data



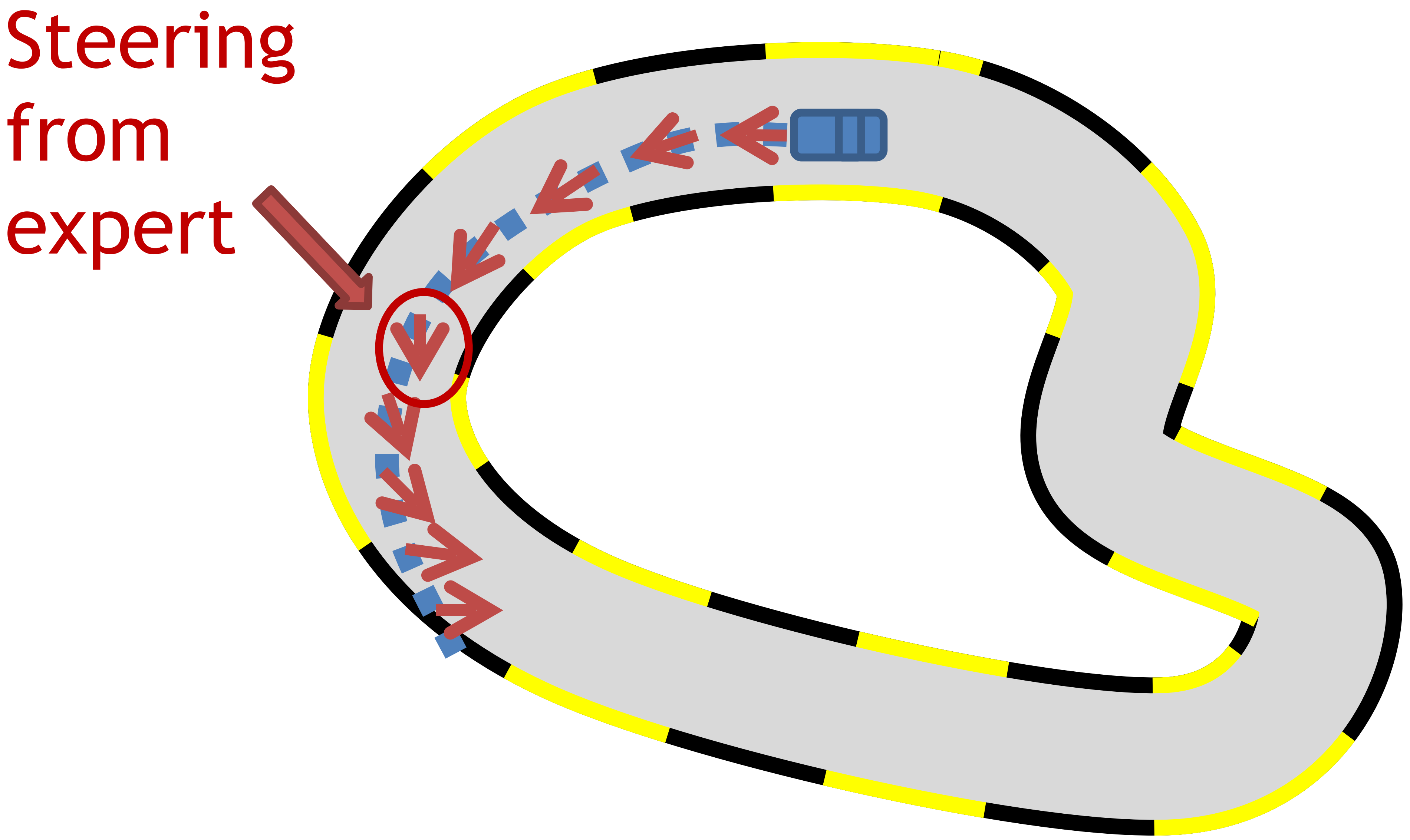
States from
the learned policy

[Ross11a]

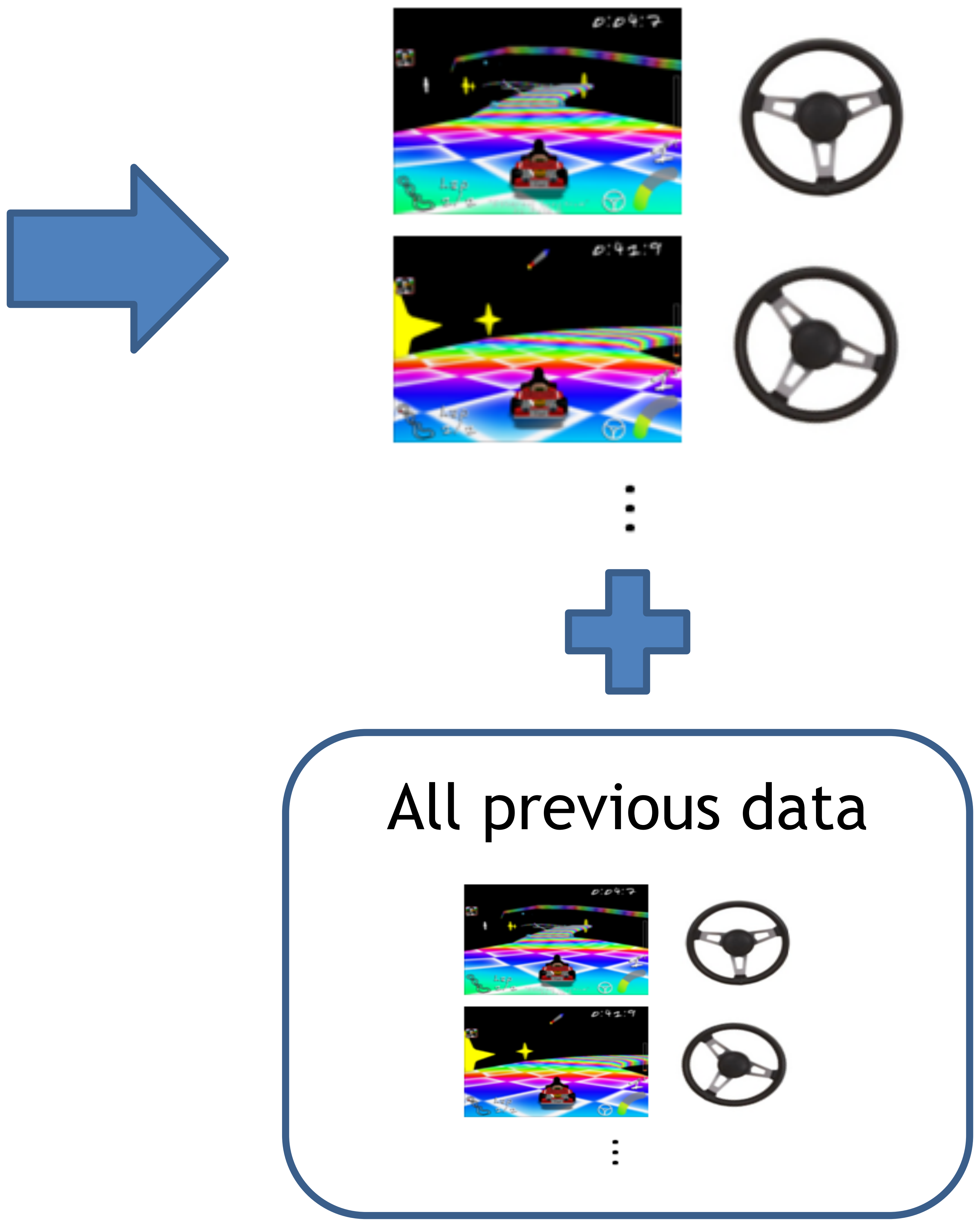
DAgger: Dataset Aggregation

1st iteration

Execute π_1 and Query Expert

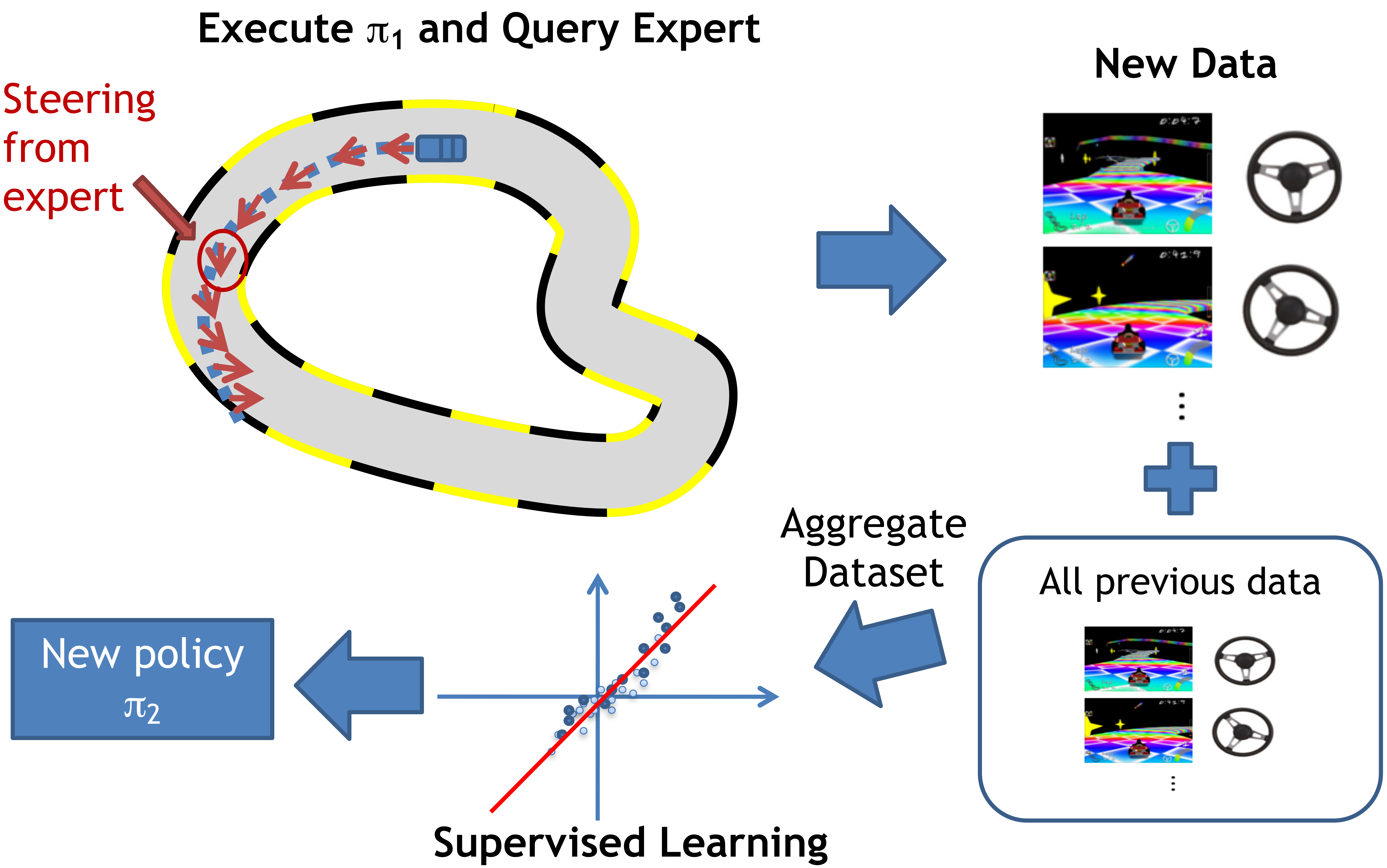


New Data



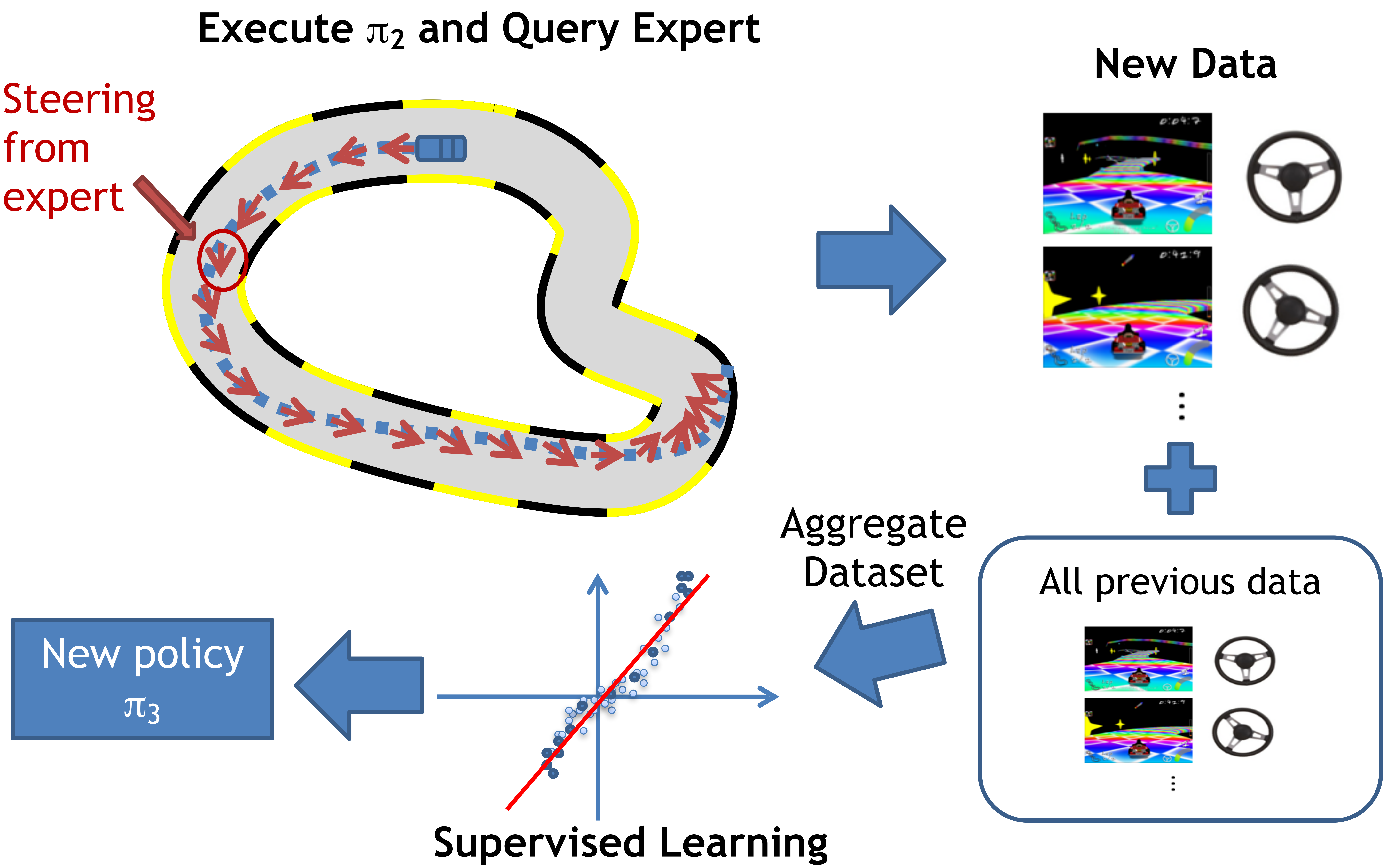
DAgger: Dataset Aggregation [Ross11a]

1st iteration



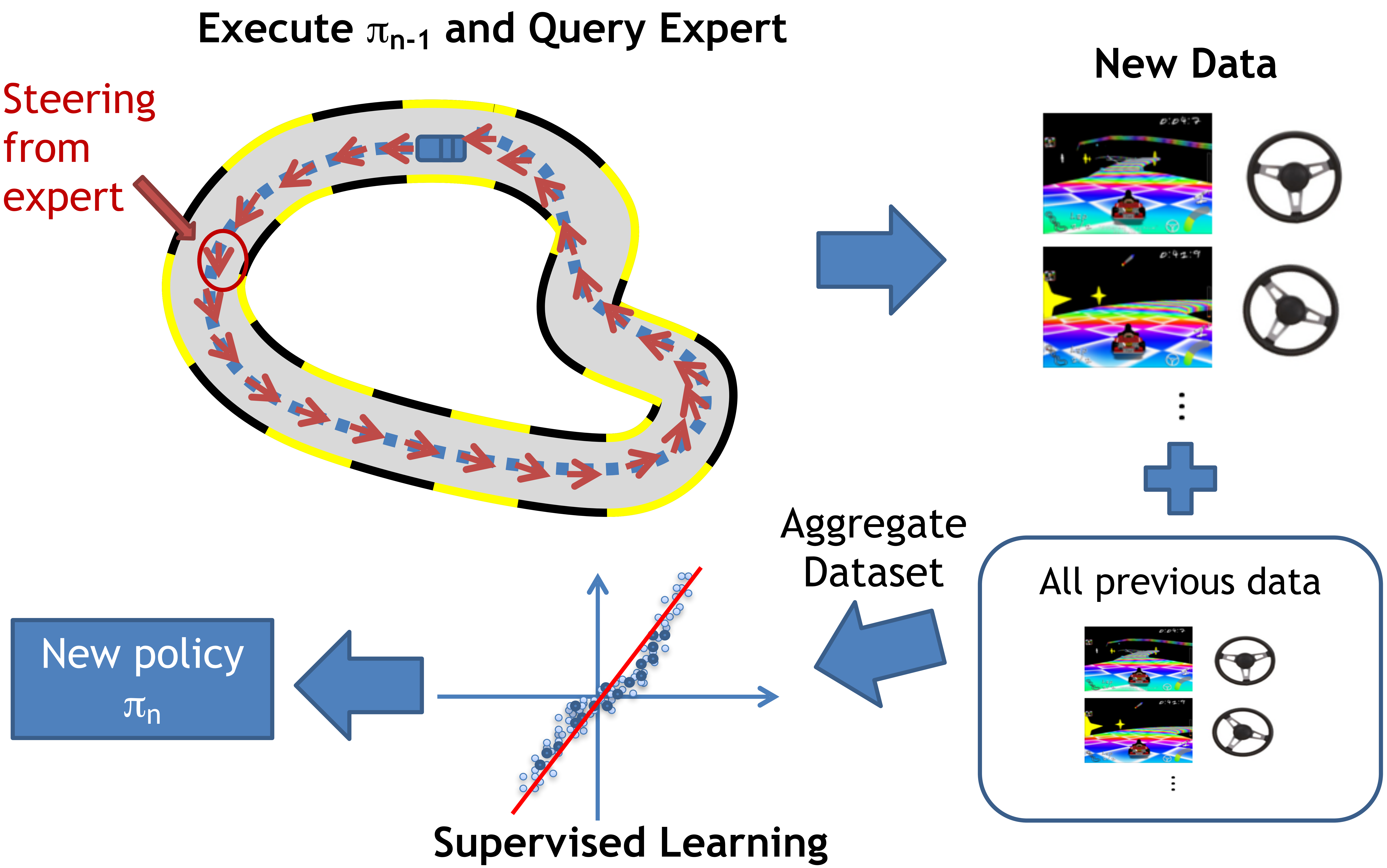
DAgger: Dataset Aggregation
[Ross11a]

2nd iteration



DAgger: Dataset Aggregation [Ross11a]

n^{th} iteration



Success!

[Ross AISTATS 2011]

Success!

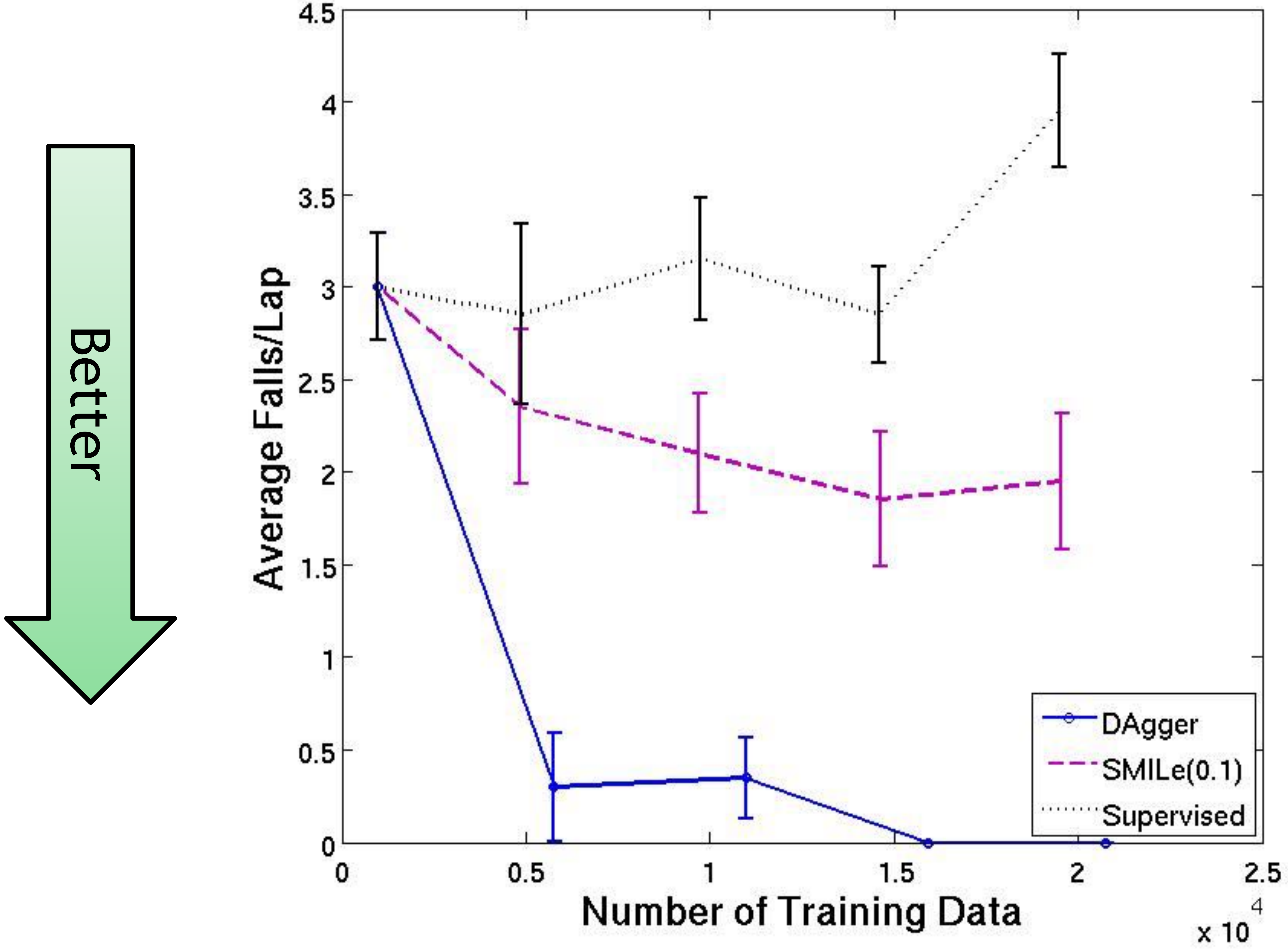
[Ross AISTATS 2011]

Success!

[Ross AISTATS 2011]

[Ross AISTATS 2011]

Average Falls/Lap



FPS: 24

Attempt: 1 of 1

AgentLinear

Selected Actions:

RIGHT SPEED

FPS: 24

Attempt: 1 of 1

AgentLinear

Selected Actions:

RIGHT SPEED

FPS: 24

Attempt: 1 of 1

AgentLinear

Selected Actions:

RIGHT SPEED

More fun than Video Games…

[Ross ICRA 2013]

More fun than Video Games…

[Ross ICRA 2013]

More fun than Video Games…

[Ross ICRA 2013]

Forms of the Interactive Experts

Interactive Expert is expensive, especially when the expert is human…

But expert does not have to be human…

Forms of the Interactive Experts

Interactive Expert is expensive, especially when the expert is human…

But expert does not have to be human…

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Fig. 4: The AutoRally car and the test track.

Forms of the Interactive Experts

Interactive Expert is expensive, especially when the expert is human…

But expert does not have to be human…

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Fig. 4: The AutoRally car and the test track.

Goal: learn a racing control policy that maps from data on cheap on-board sensors (raw-pixel imagine) to low-level control (steer and throttle)

Forms of the Interactive Experts

Interactive Expert is expensive, especially when the expert is human…

But expert does not have to be human…

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Fig. 4: The AutoRally car and the test track.

Goal: learn a racing control policy that maps from data on cheap on-board sensors (raw-pixel imagine) to low-level control (steer and throttle)

(a) raw image

→ Steering + throttle

Forms of the Interactive Experts

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Forms of the Interactive Experts

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Their Setup:
At Training, we have expensive sensors for accurate state estimation
and we have computation resources for **MPC** (i.e., high-frequency replanning)

Forms of the Interactive Experts

Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Their Setup:
At Training, we have expensive sensors for accurate state estimation
and we have computation resources for **MPC** (i.e., high-frequency replanning)

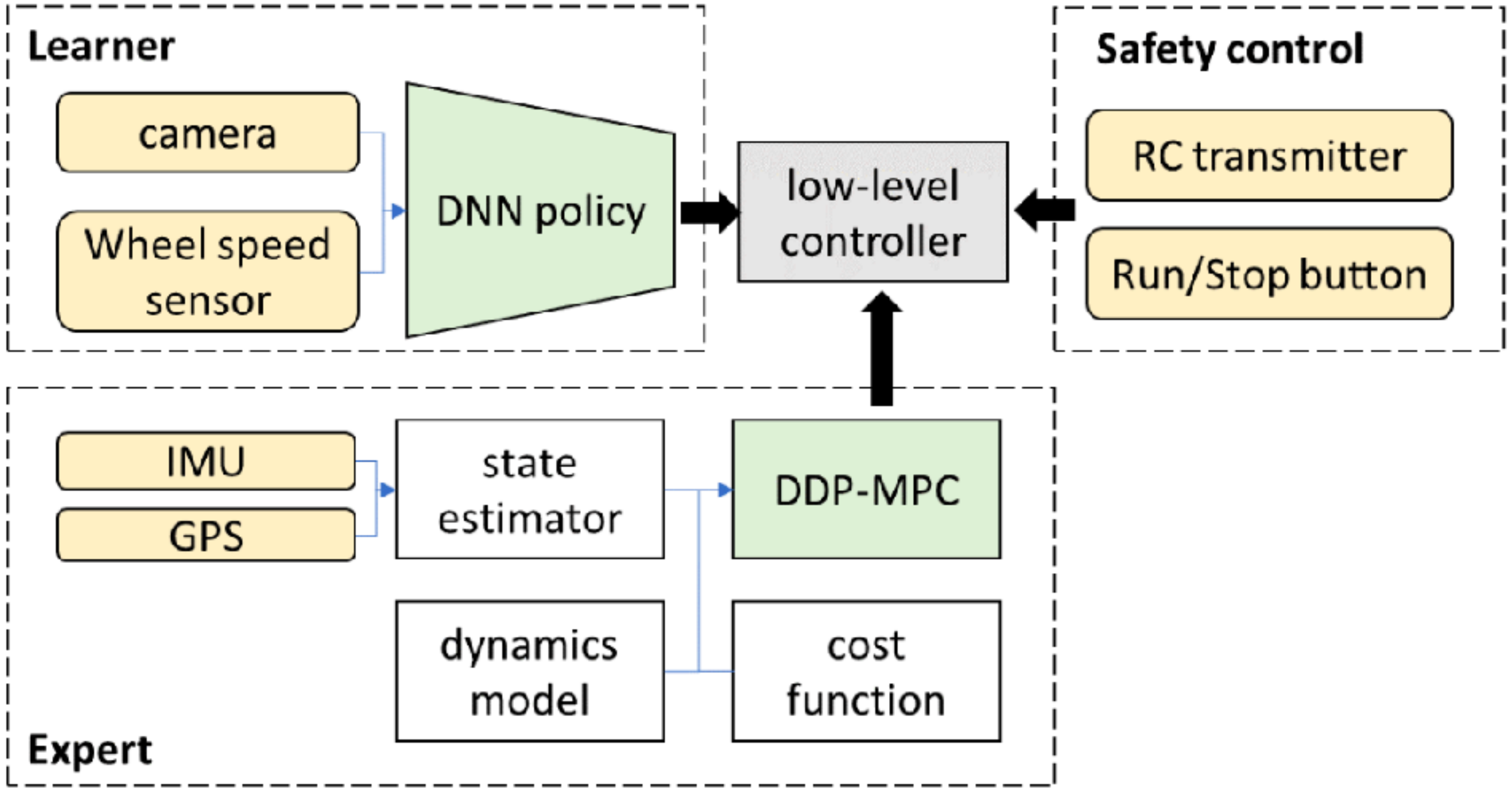
The MPC is the expert in this case!

Forms of the Interactive Experts

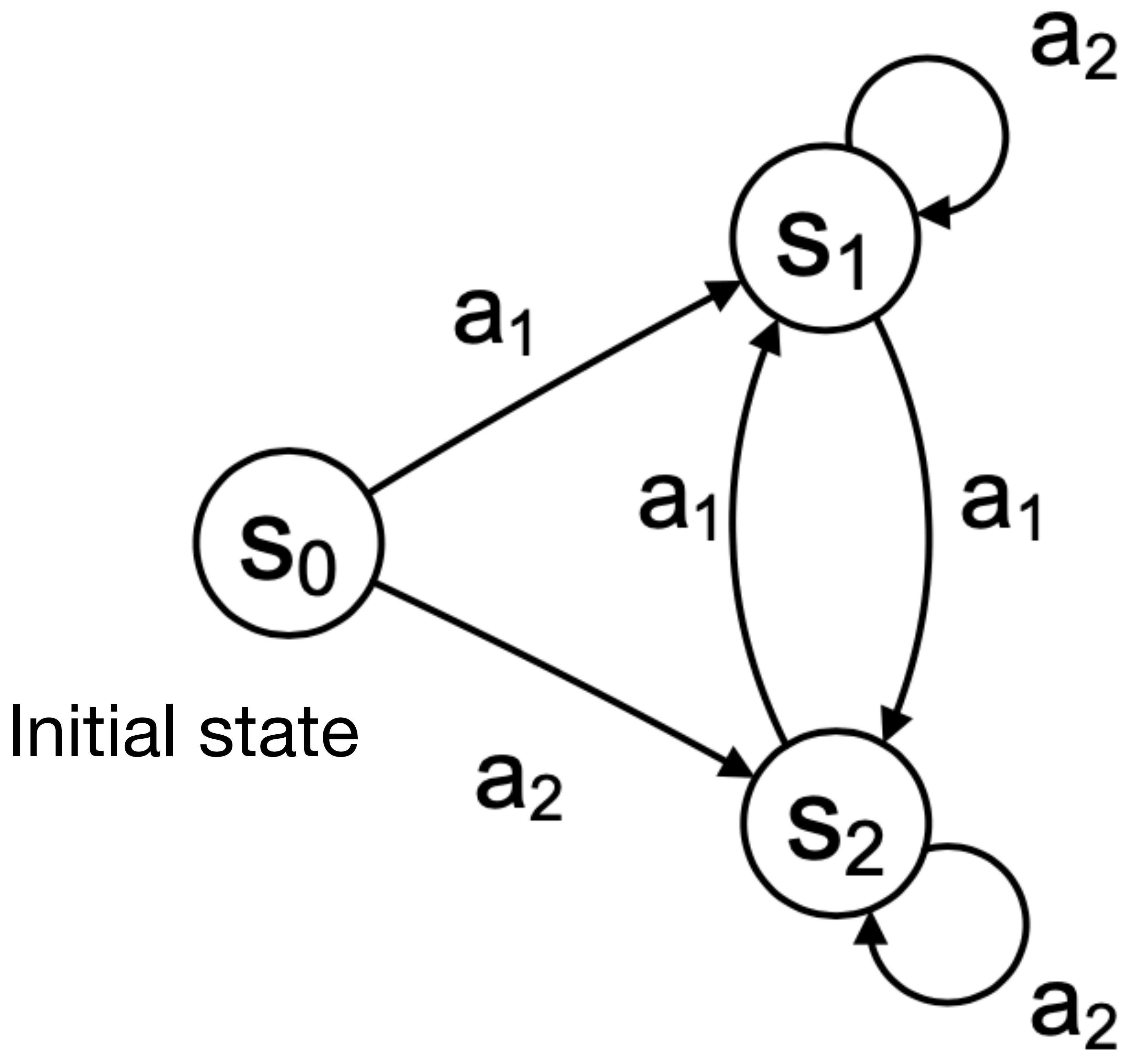
Example: high-speed off-road driving
[Pan et al, RSS 18, Best System Paper]

Their Setup:
At Training, we have expensive sensors for accurate state estimation
and we have computation resources for **MPC** (i.e., high-frequency replanning)

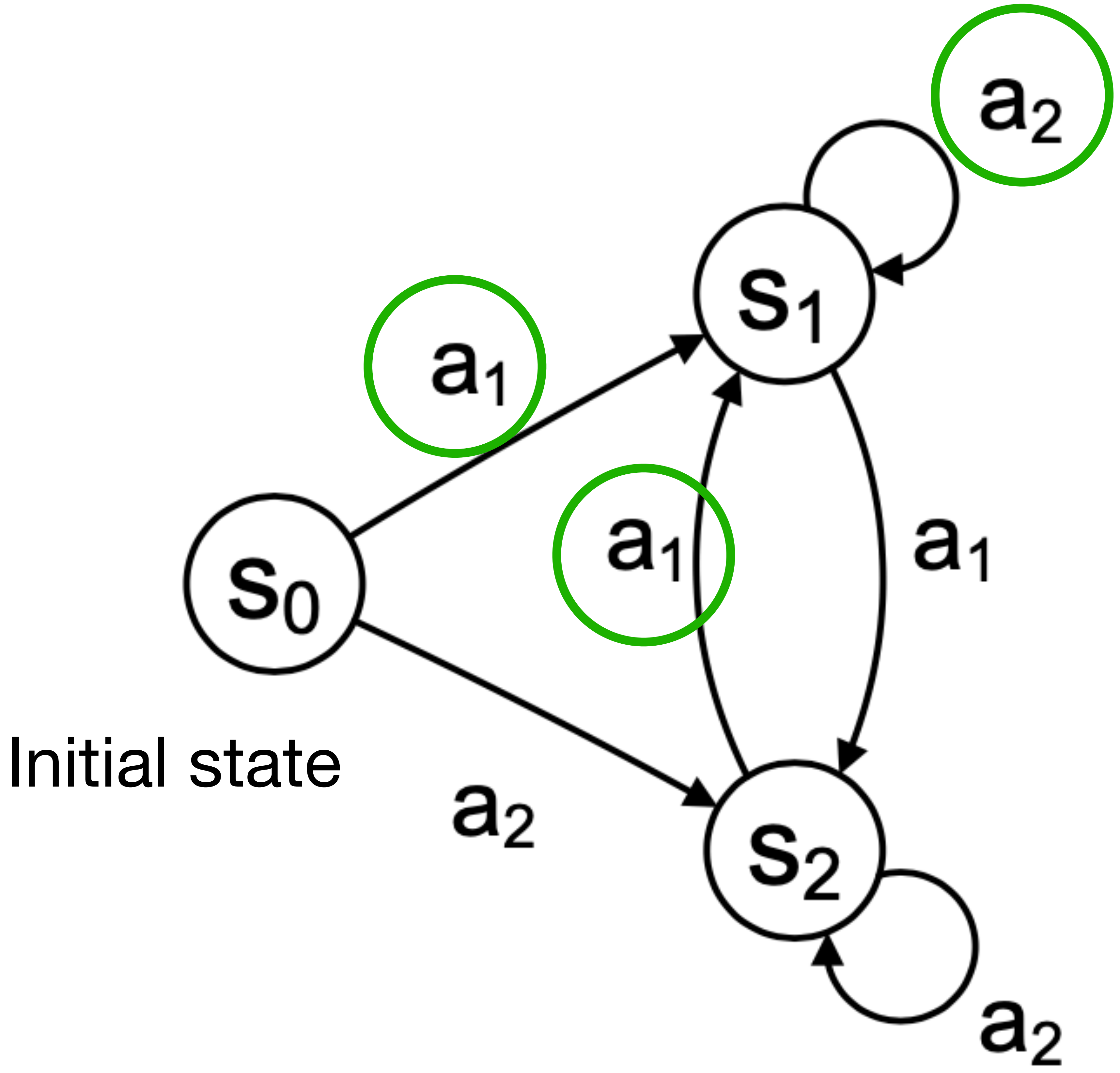
The MPC is the expert in this case!



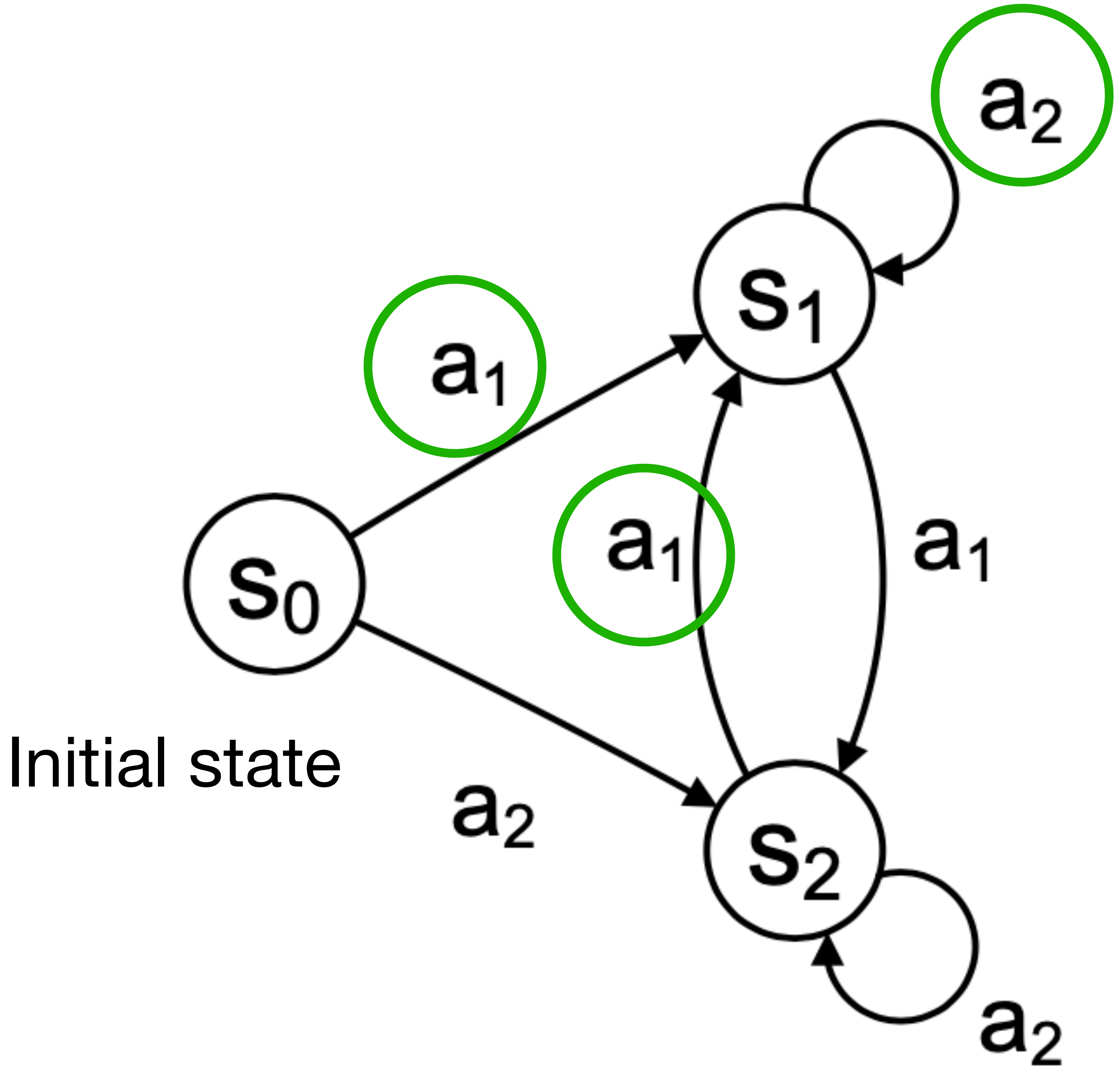
Distribution Shift: Example



Distribution Shift: Example



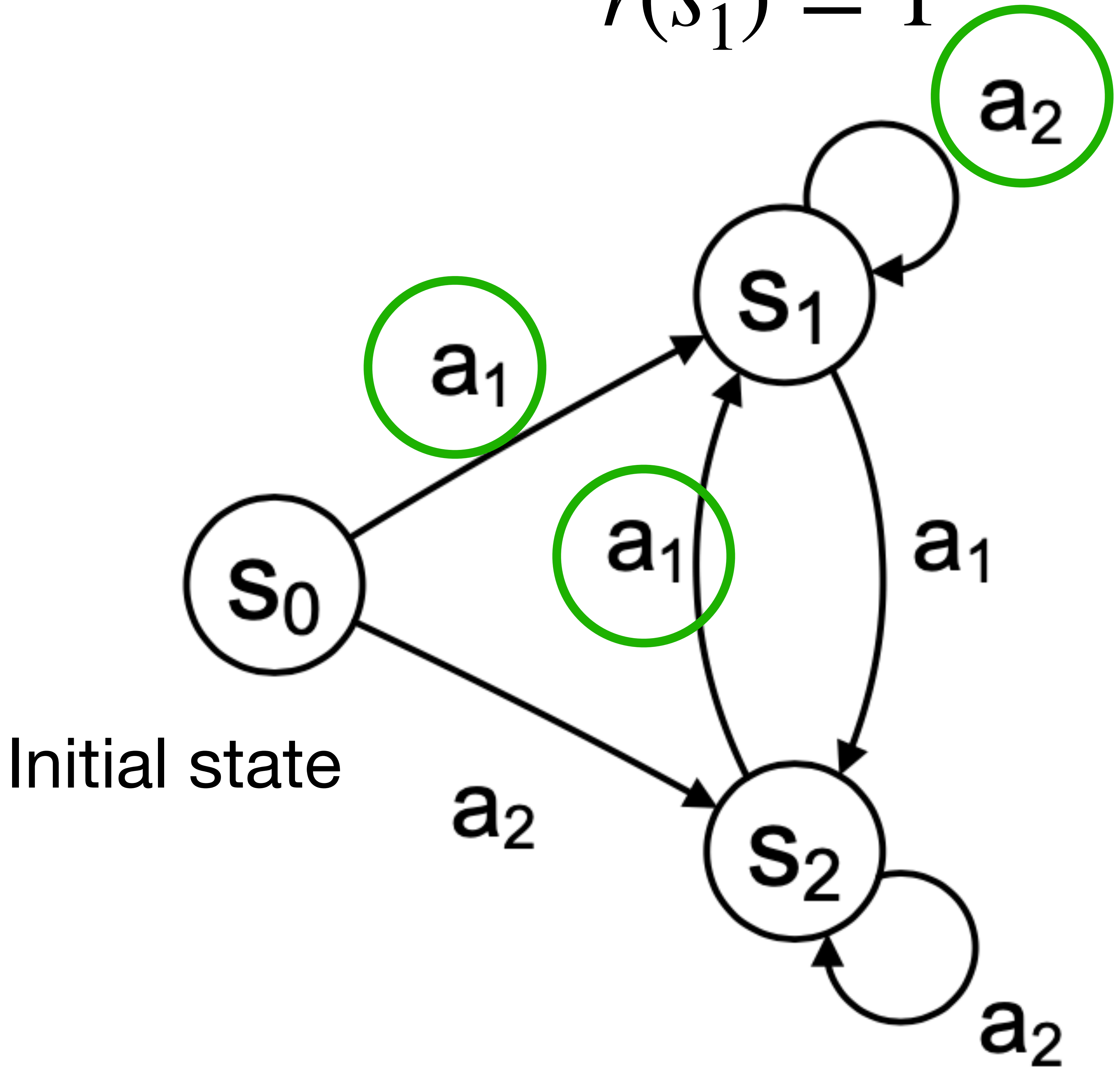
Distribution Shift: Example



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

Distribution Shift: Example

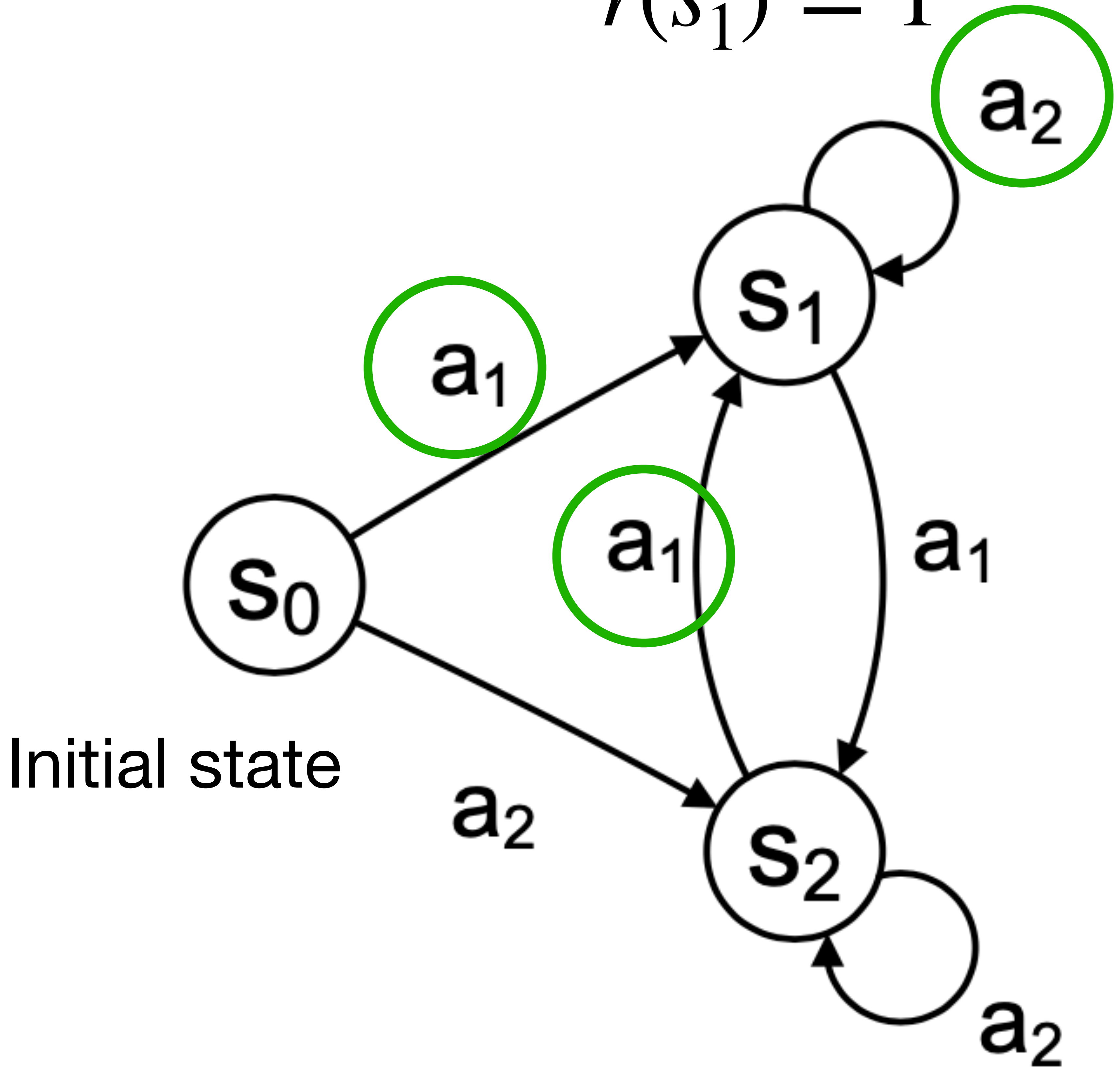
$r(s_1) = 1$



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

Distribution Shift: Example

$r(s_1) = 1$

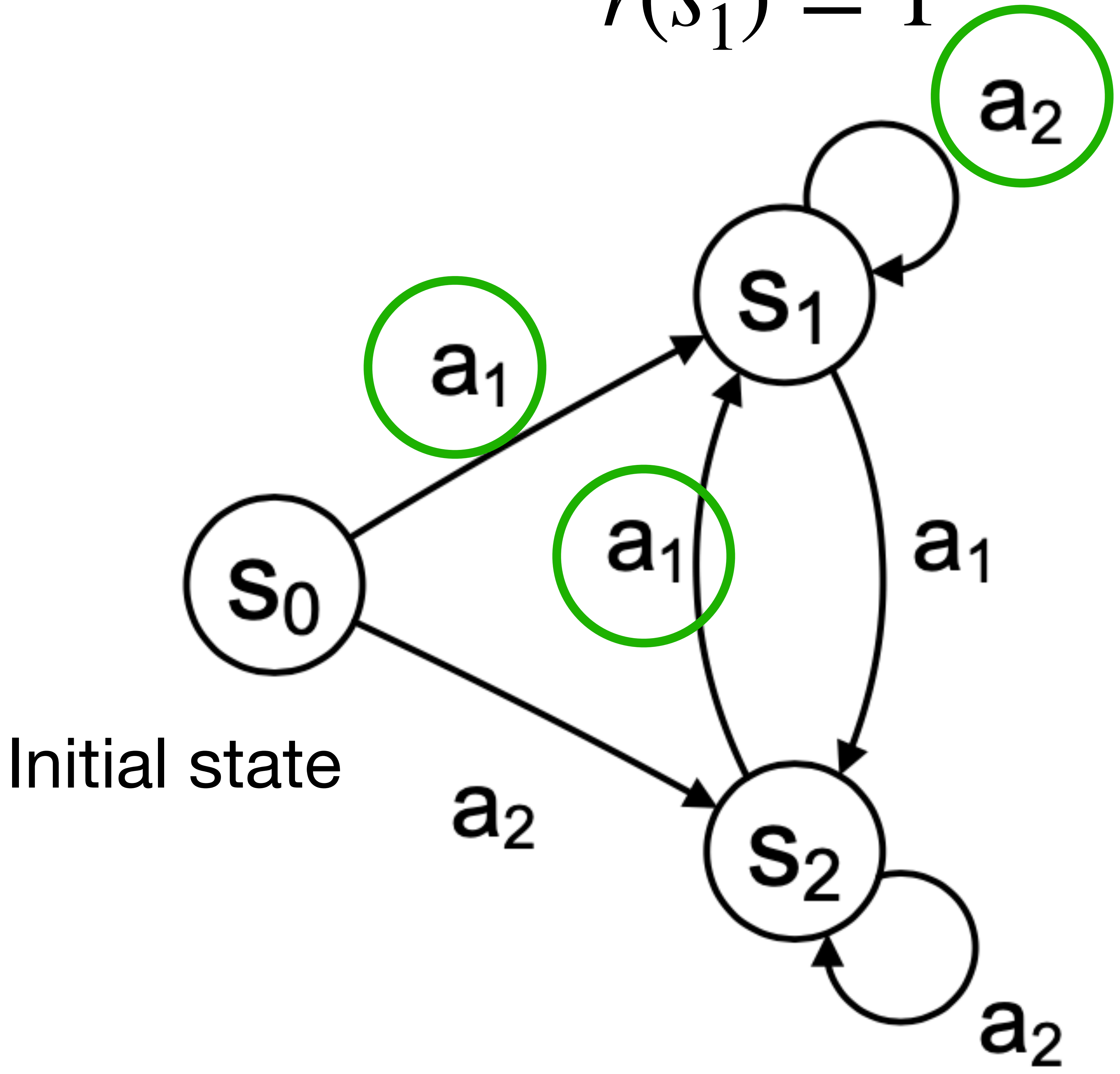


$$d_{s_0}^{\pi^\star}(s_0) = 1 - \gamma,\ d_{s_0}^{\pi^\star}(s_1) = \gamma,\ d_{s_0}^{\pi^\star}(s_2) = 0$$

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

Distribution Shift: Example

$r(s_1) = 1$



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

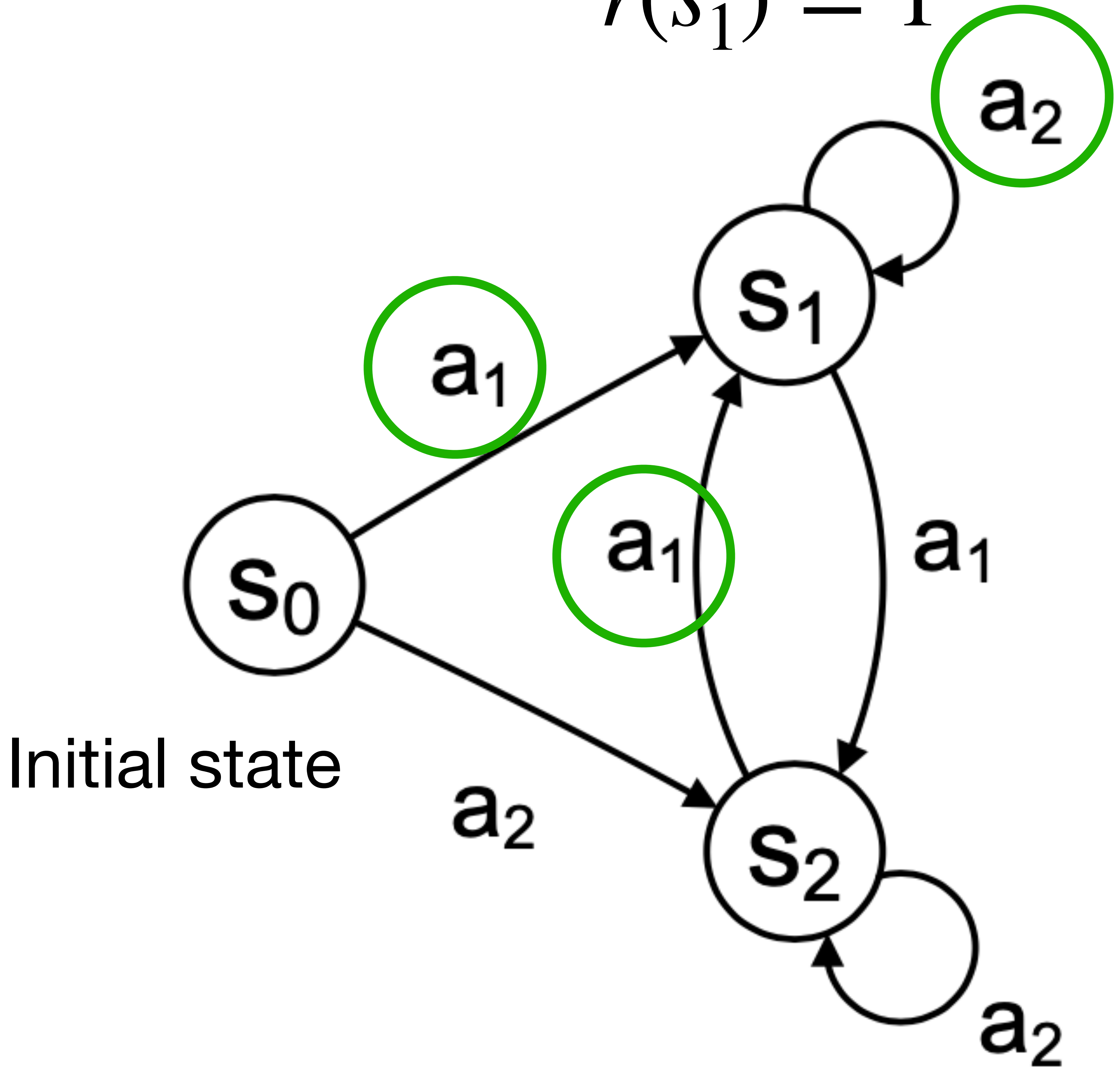
$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

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

$r(s_1) = 1$



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

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

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

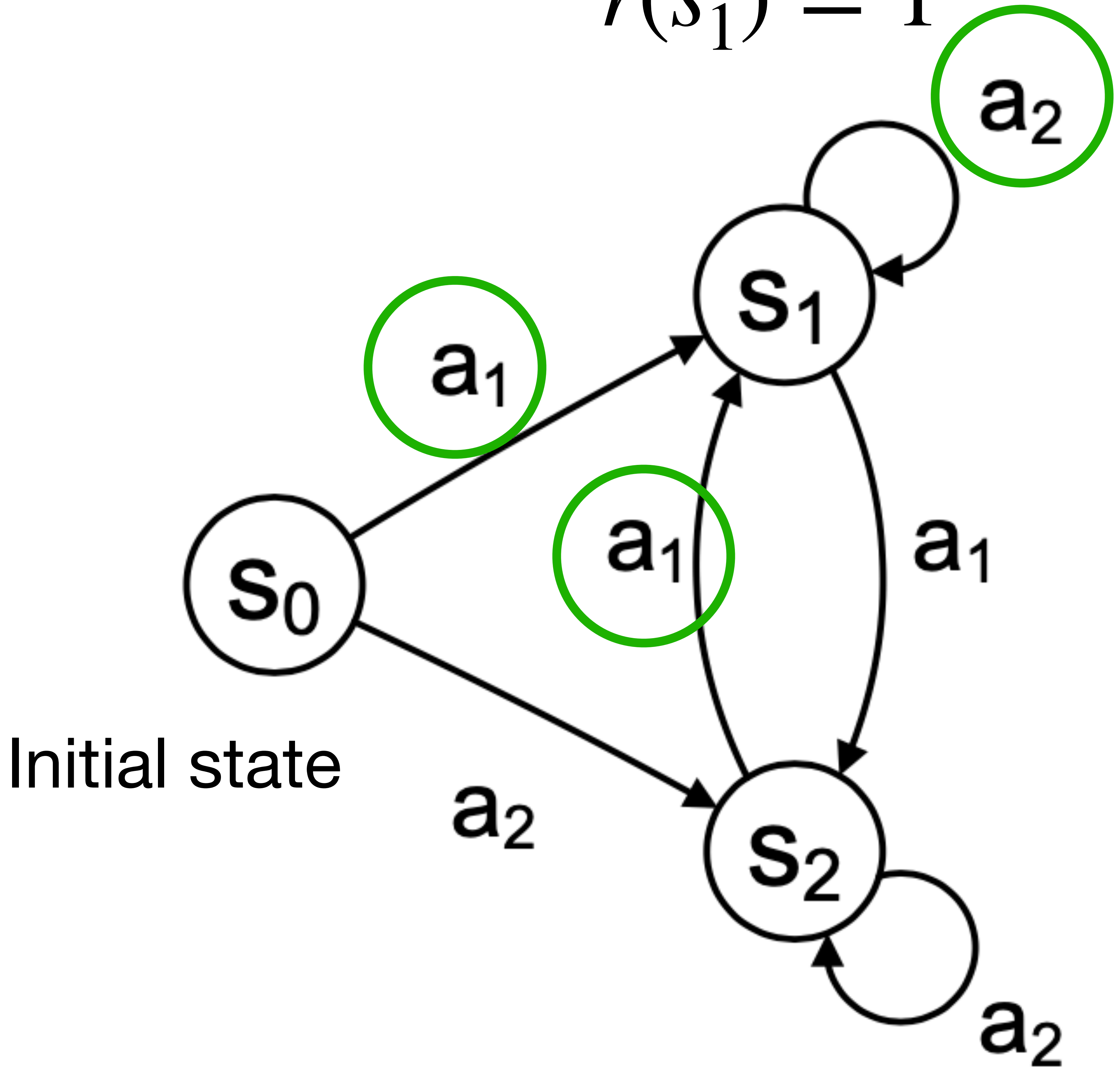
$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1-\epsilon/(1-\gamma) \\ a_2 & \text{w/ prob } \epsilon/(1-\gamma) \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

We will have good supervised learning error:

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

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:

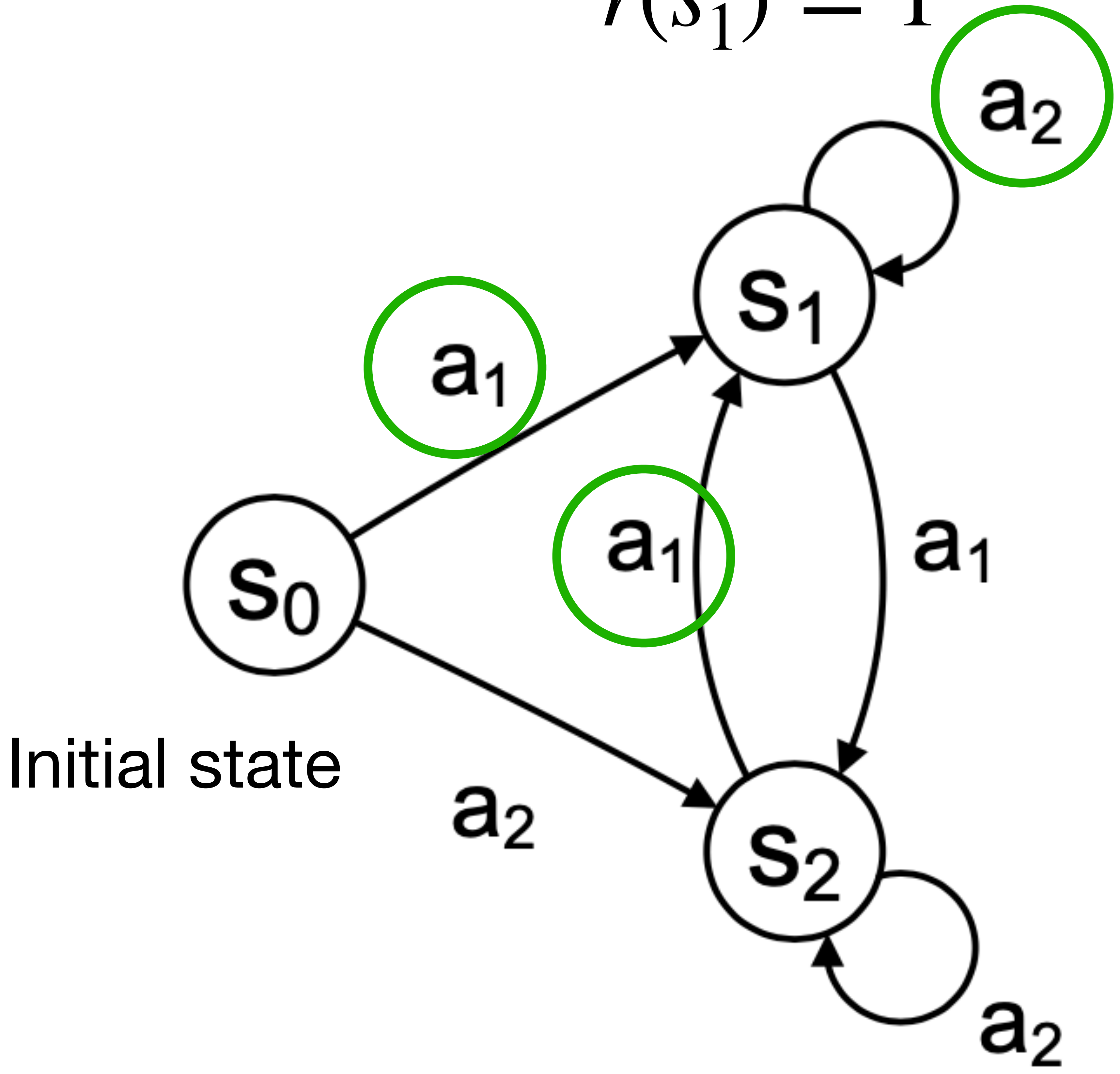
$$\mathbb{E}_{s \sim d_{s_0}^{\pi^\star}} \mathbb{E}_{a \sim \widehat{\pi}(\cdot|s)} \mathbf{1}\left(a \neq \pi^\star(s)\right) = \epsilon$$

But we have quadratic error in performance:

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

Distribution Shift: Example

$$r(s_1) = 1$$



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

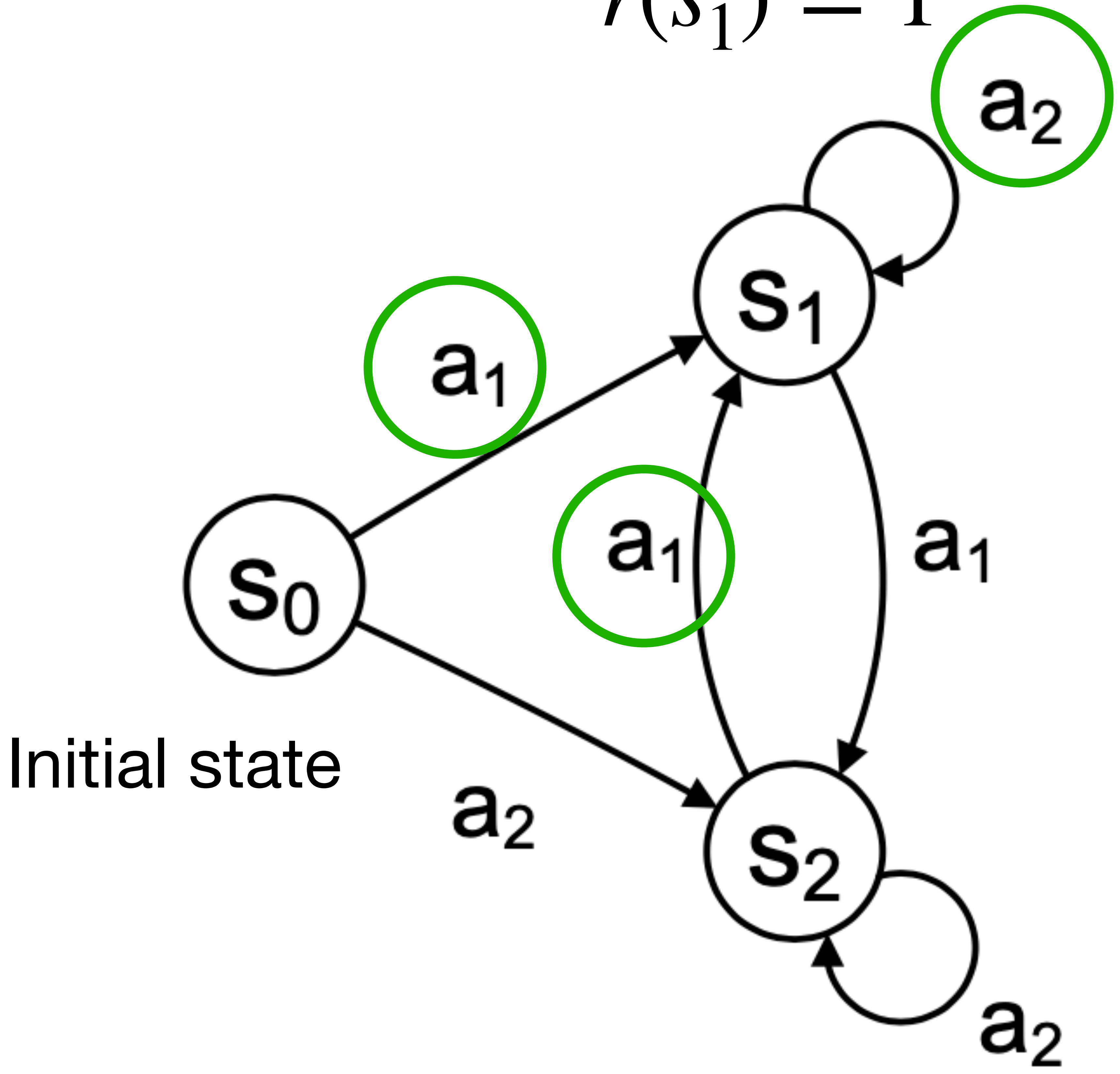
$$V_{s_0}^{\pi^\star} = \frac{\gamma}{1-\gamma}$$

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

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

Distribution Shift: Example

$r(s_1) = 1$



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

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

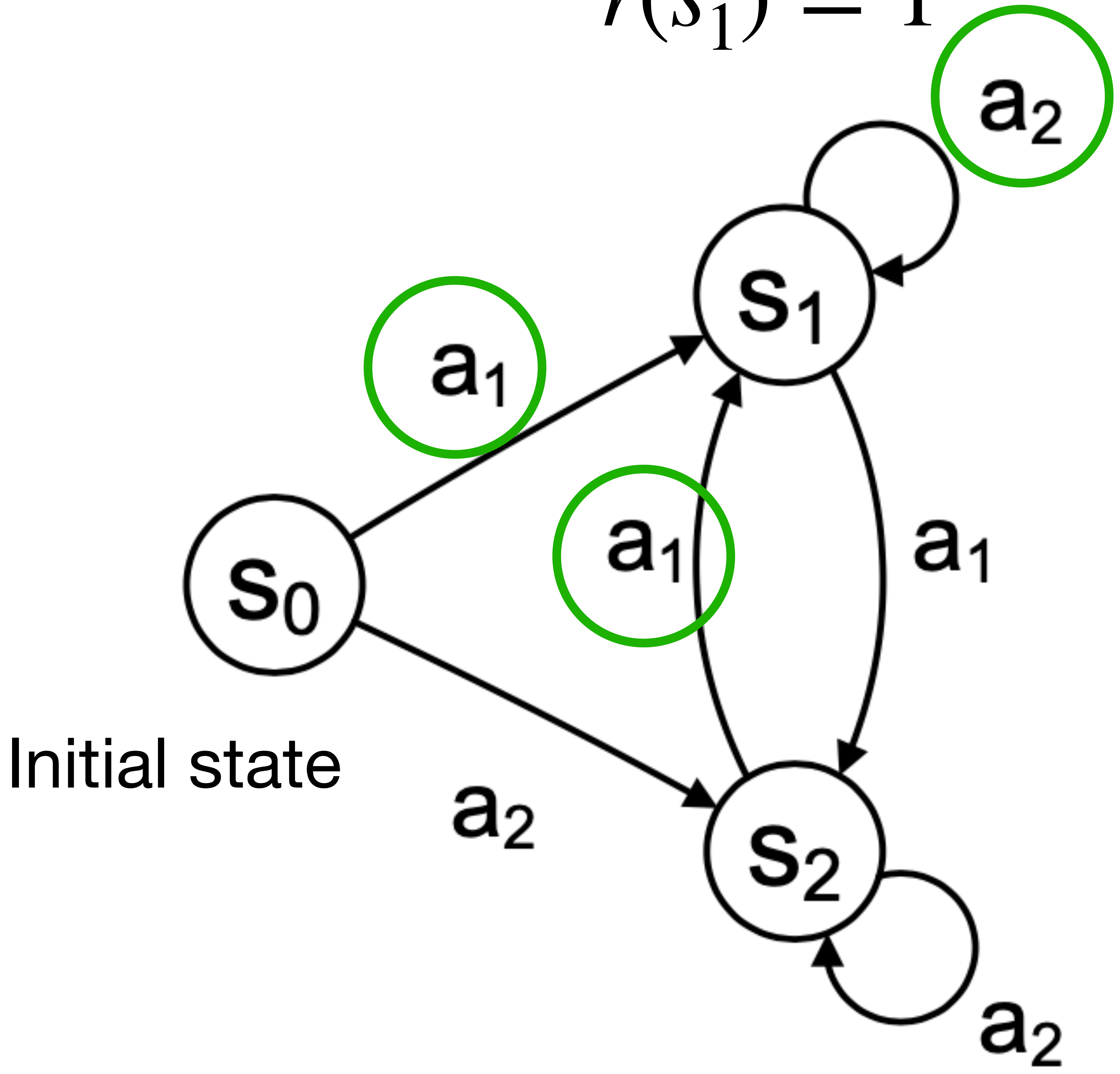
Assume SL returned such policy $\widehat{\pi}$

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

Why DAgger can fix this problem?

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 $\widehat{\pi}$

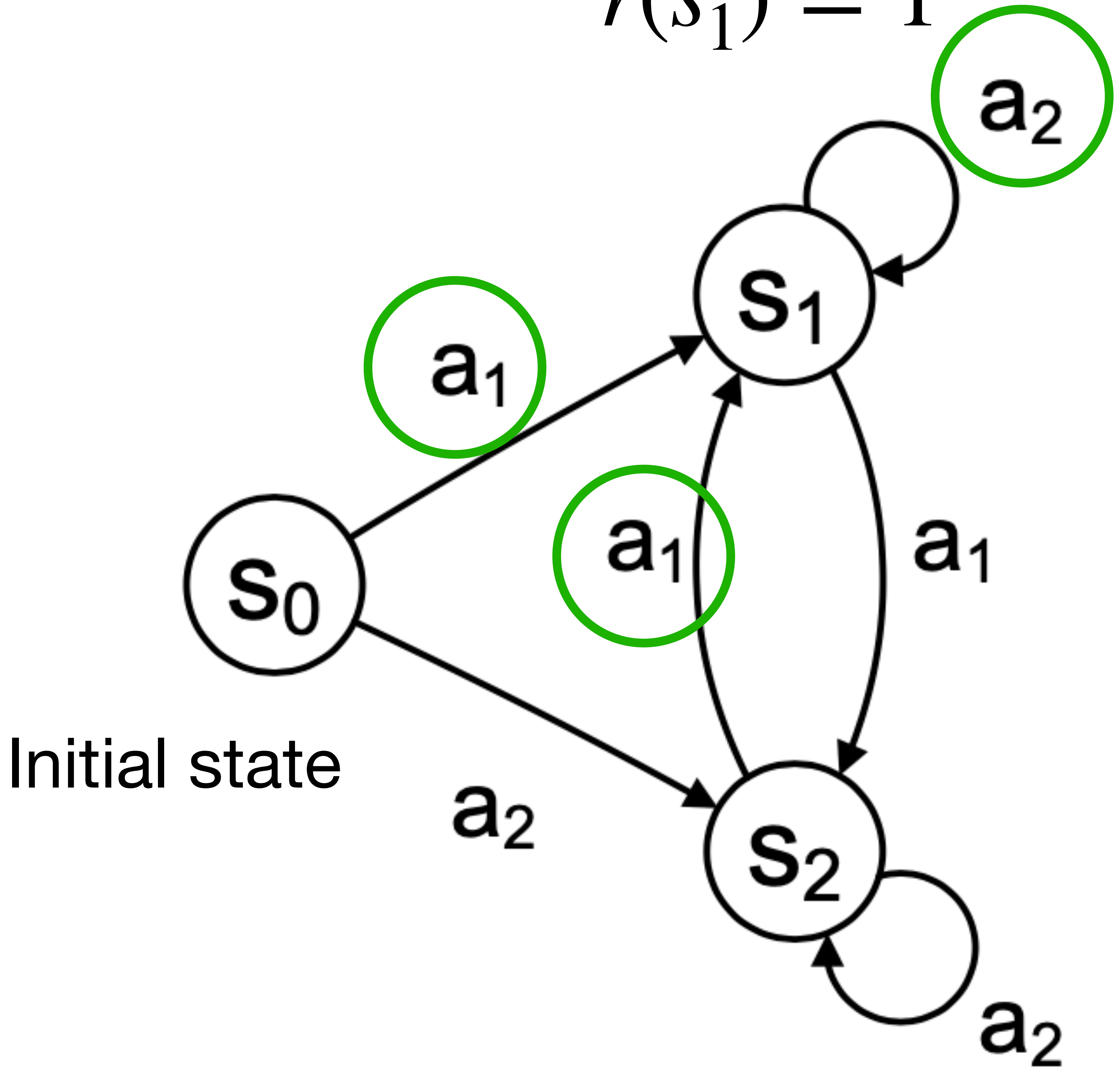
$$\widehat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1-\epsilon/(1-\gamma) \\ a_2 & \text{w/ prob } \epsilon/(1-\gamma) \end{cases}, \quad \widehat{\pi}(s_1) = a_2, \widehat{\pi}(s_2) = a_2$$

Why DAgger can fix this problem?

$\widehat{\pi}$ will visit s_2, and we collect $\pi^\star(s_2) = a_1$

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 $\widehat{\pi}$

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

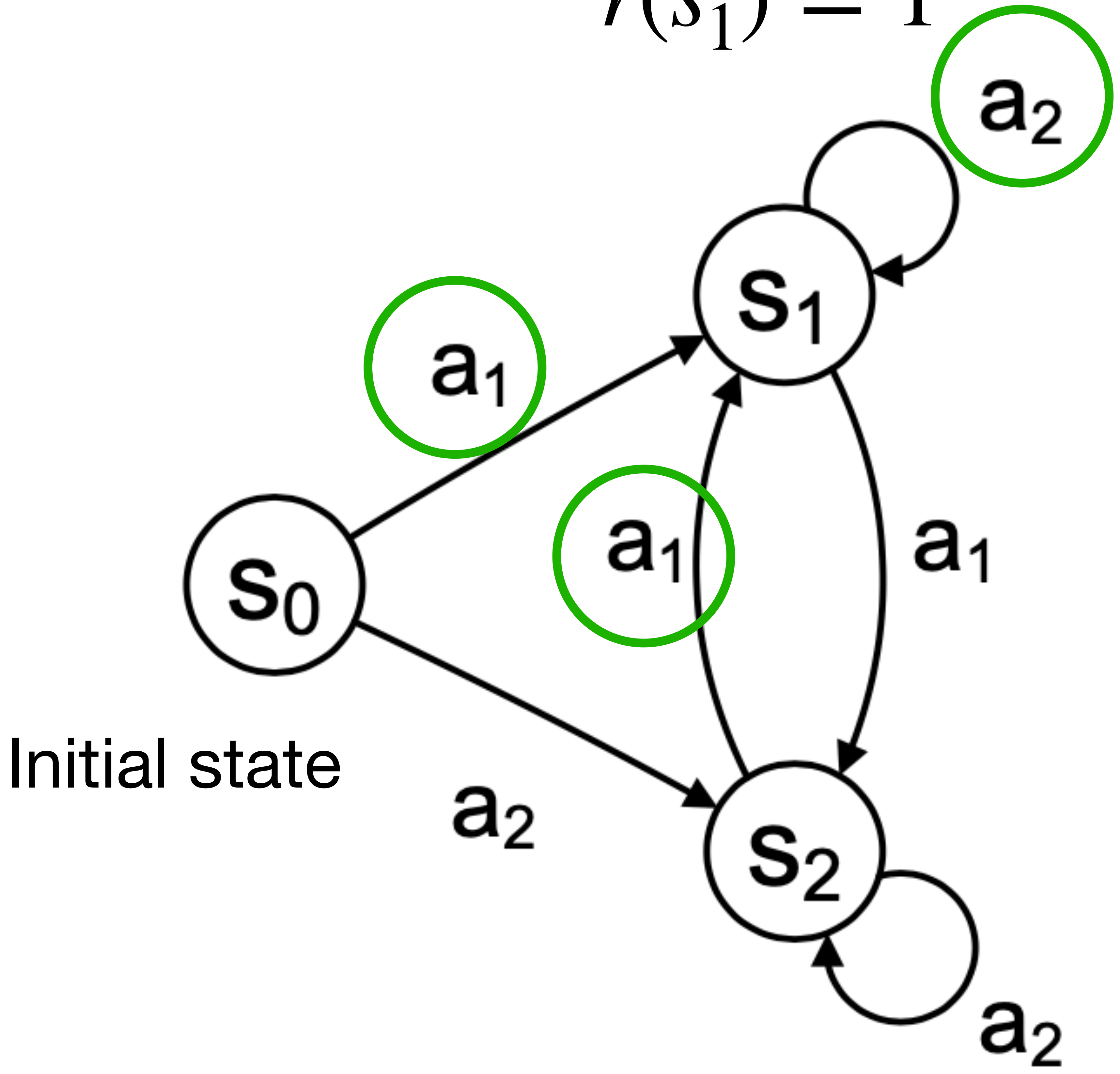
Why DAgger can fix this problem?

$\widehat{\pi}$ will visit s_2, and we collect $\pi^\star(s_2) = a_1$

By data aggregation, our new dataset will contain (s_2, a_1) pairs

Distribution Shift: Example

$r(s_1) = 1$



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

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

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

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

Why DAgger can fix this problem?

$\widehat{\pi}$ will visit s_2, and we collect $\pi^\star(s_2) = a_1$

By data aggregation, our new dataset will contain (s_2, a_1) pairs

Thus, our new learned policy will know what to do at s_2

Outline for today:

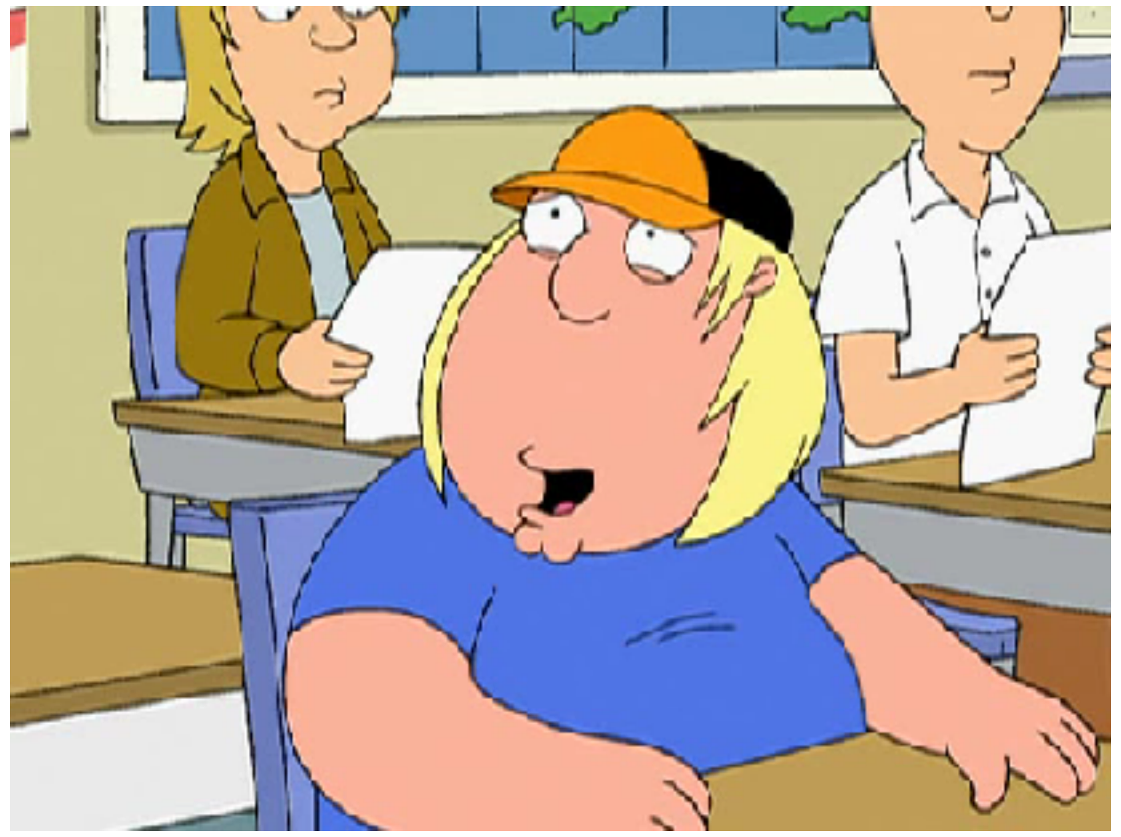
1. The DAgger (Data Aggregation) Algorithm

2. Quick intro on Online Learning

[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner



convex Decision set Θ

Adversary

...

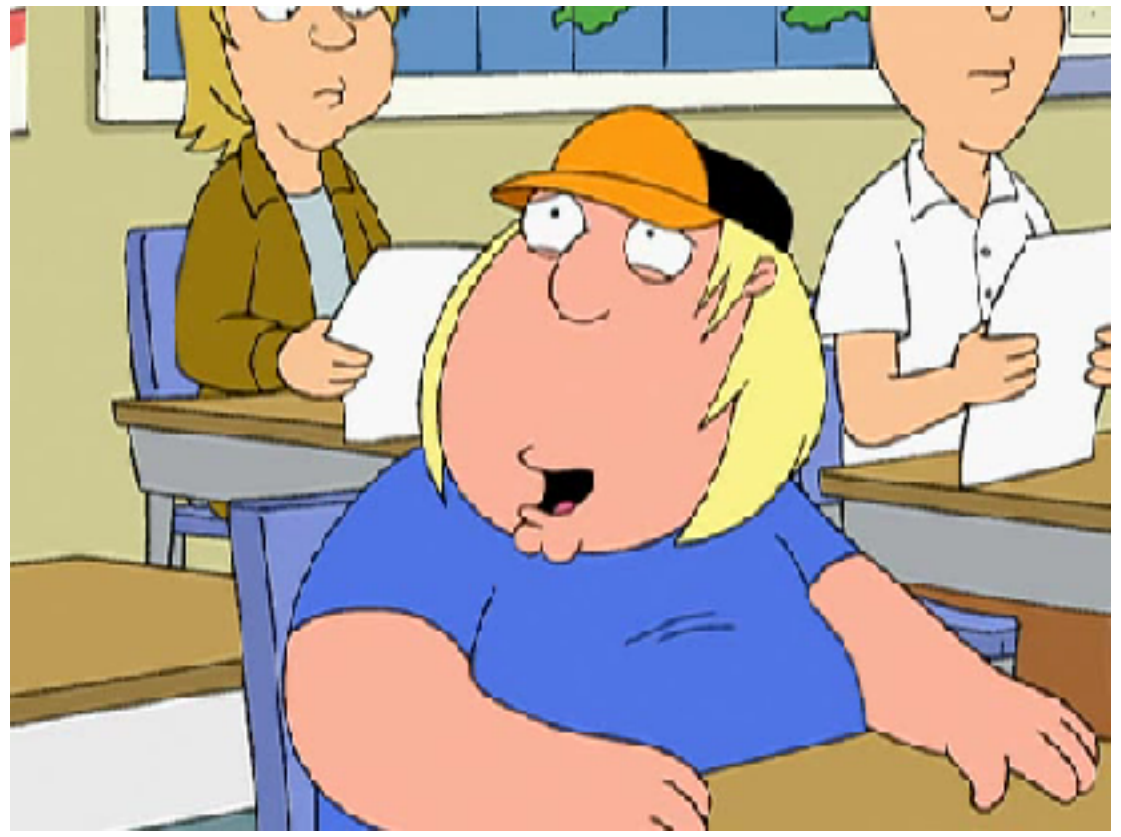
[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner

Learner picks a decision θ_0

Adversary



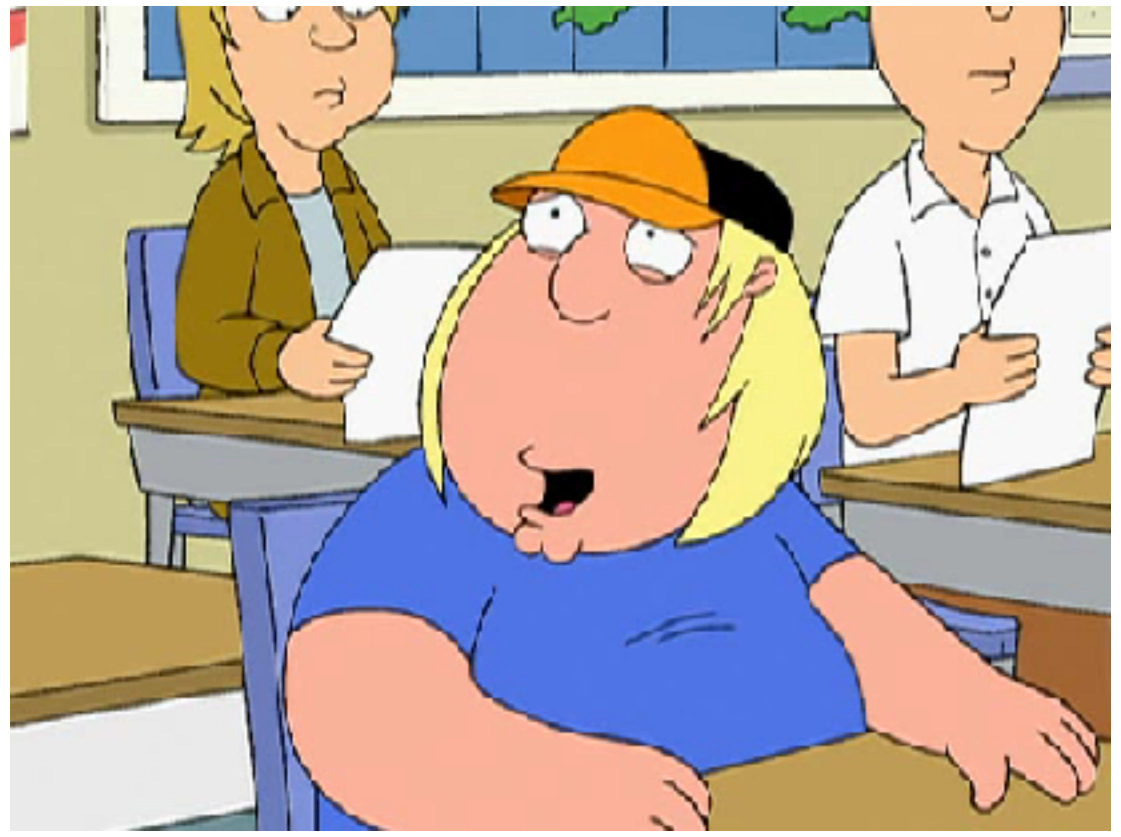
convex Decision set Θ

...

[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner



convex Decision set Θ

Learner picks a decision θ_0

Adversary picks a loss $\ell_0 : \Theta \to \mathbb{R}$

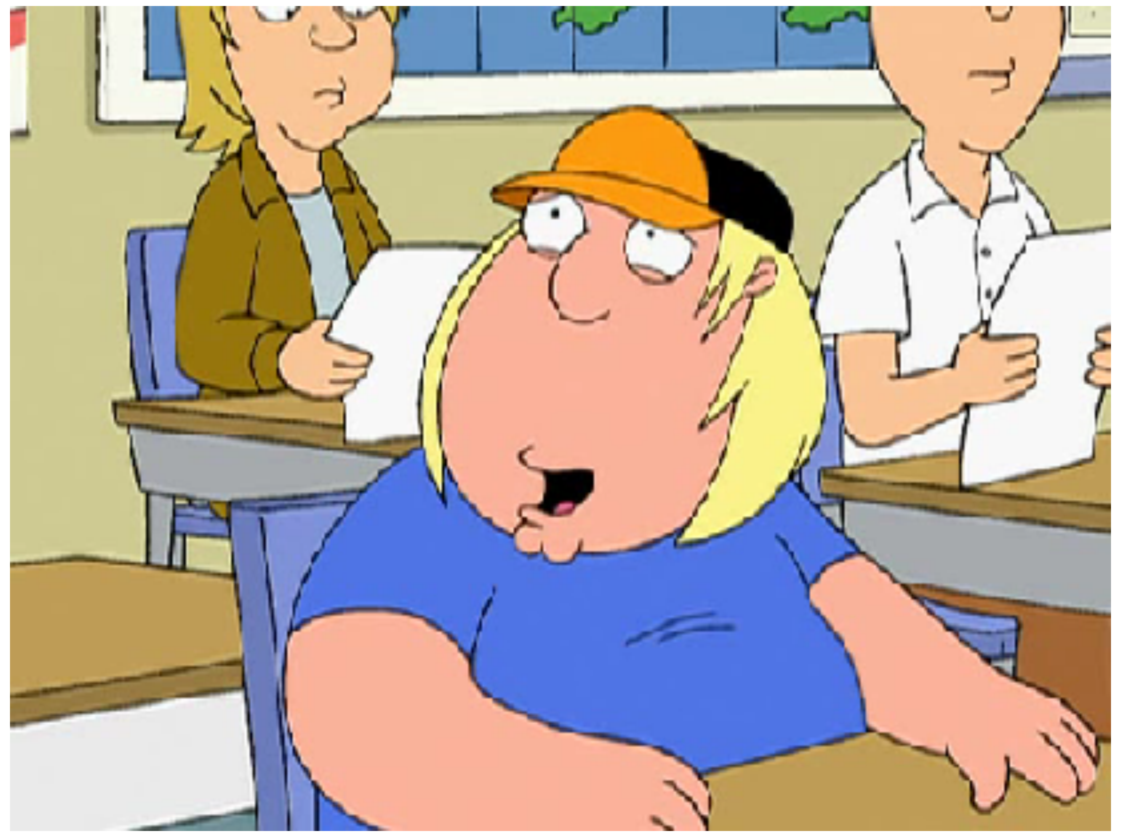
...

Adversary

[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner



convex Decision set Θ

Learner picks a decision θ_0 →

← Adversary picks a loss $\ell_0 : \Theta \to \mathbb{R}$

Learner picks a new decision θ_1 →

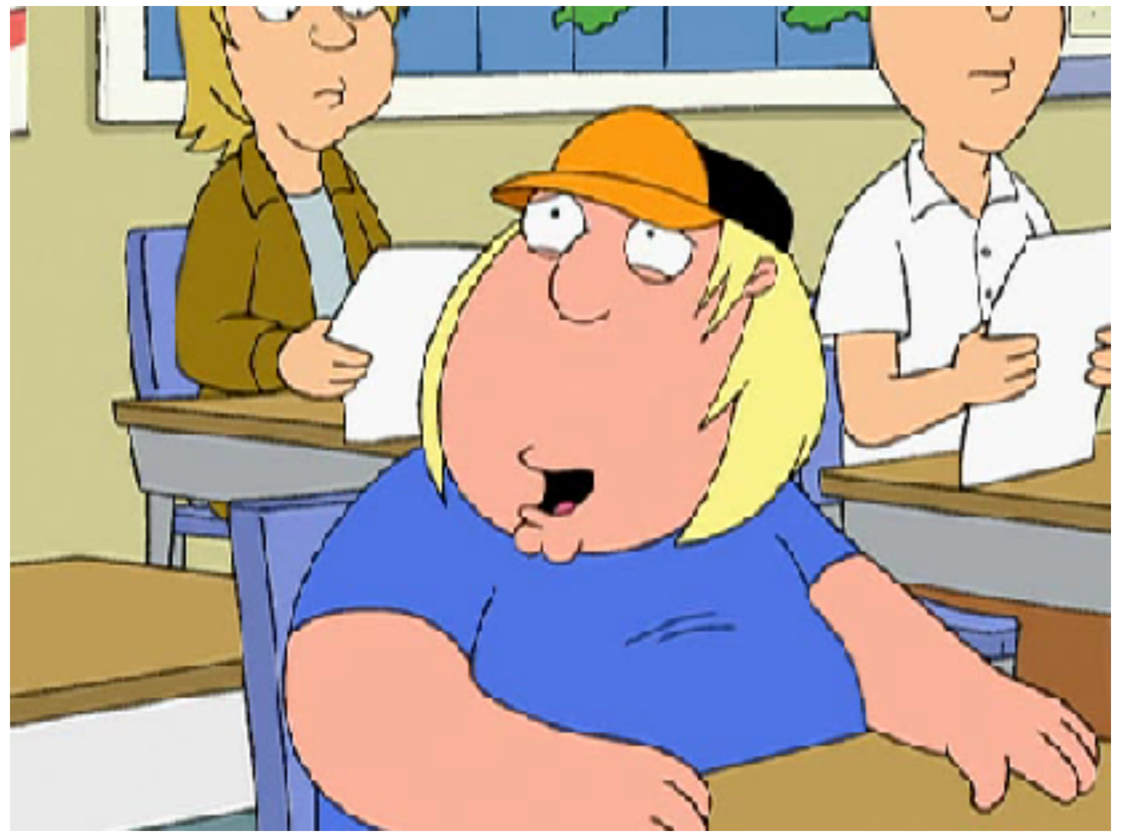
...

Adversary

[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner



convex Decision set Θ

Learner picks a decision θ_0 $\longrightarrow$

Adversary picks a loss $\ell_0 : \Theta \to \mathbb{R}$ $\longleftarrow$

Learner picks a new decision θ_1 $\longrightarrow$

Adversary picks a loss $\ell_1 : \Theta \to \mathbb{R}$ $\longleftarrow$

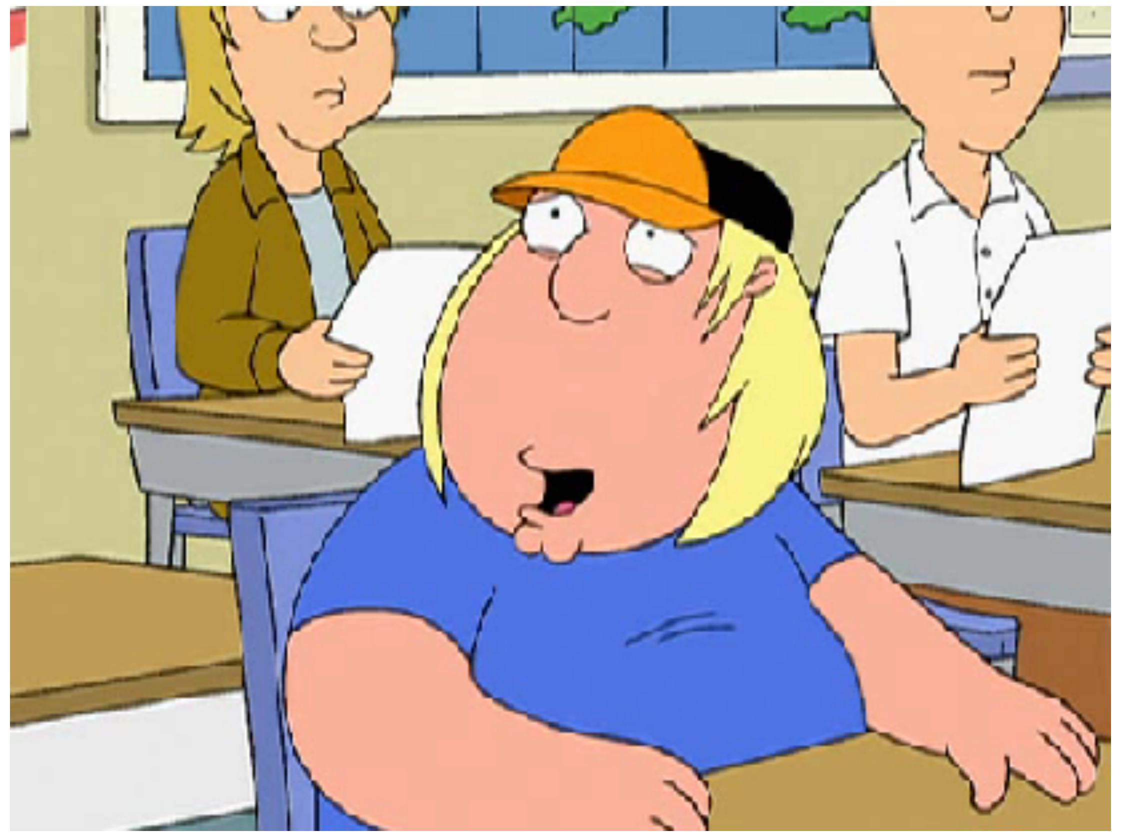
...

Adversary

[Vovk92,Warmuth94,Freund97,Zinkevich03,Kalai05,Hazan06,Kakade08]

Online Learning

Learner



convex Decision set Θ

Learner picks a decision θ_0 →

← Adversary picks a loss $\ell_0 : \Theta \to \mathbb{R}$

Learner picks a new decision θ_1 →

← Adversary picks a loss $\ell_1 : \Theta \to \mathbb{R}$

...

Adversary

$$\text{Regret} = \sum_{t=0}^{T-1} \ell_t(\theta_t) - \min_{\theta \in \Theta} \sum_{t=0}^{T-1} \ell_t(\theta)$$

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

1. Learner first picks $\theta_t \in \text{Ball} \subset \mathbb{R}^d$

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

1. Learner first picks $\theta_t \in$ Ball $\subset \mathbb{R}^d$

2. Adversary **then** picks $x_t \in \mathcal{X} \subset \mathbb{R}^d, y_t \in [a, b]$

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

1. Learner first picks $\theta_t \in \text{Ball} \subset \mathbb{R}^d$

2. Adversary **then** picks $x_t \in \mathscr{X} \subset \mathbb{R}^d, y_t \in [a, b]$

3. Learner suffers loss $\ell_t(\theta_t) = (\theta_t^\top x_t - y_t)^2$

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

1. Learner first picks $\theta_t \in \text{Ball} \subset \mathbb{R}^d$

2. Adversary **then** picks $x_t \in \mathscr{X} \subset \mathbb{R}^d, y_t \in [a, b]$

3. Learner suffers loss $\ell_t(\theta_t) = (\theta_t^\top x_t - y_t)^2$

Learner has to make decision θ_t based on history up to $t - 1$,
while adversary could pick (x_t, y_t) even after seeing θ_t

Example: online linear regression

Can we perform linear regression in online fashion with non i.i.d (or even adversary) data?

Every iteration t :

1. Learner first picks $\theta_t \in$ Ball $\subset \mathbb{R}^d$

2. Adversary **then** picks $x_t \in \mathcal{X} \subset \mathbb{R}^d, y_t \in [a, b]$

3. Learner suffers loss $\ell_t(\theta_t) = (\theta_t^\top x_t - y_t)^2$

Learner has to make decision θ_t based on history up to $t - 1$,
while adversary could pick (x_t, y_t) even after seeing θ_t

Adversary seems too powerful…

Example: online linear regression

BUT, a very intuitive algorithm actually achieves no-regret property:

Example: online linear regression

BUT, a very intuitive algorithm actually achieves no-regret property:

Every iteration t :

1. Learner first picks θ_t that minimizes the aggregated loss

$$\theta_t = \arg \min_{\theta \in \text{Ball}} \sum_{i=0}^{t-1} \left(\theta^\top x_i - y_i\right)^2 + \lambda \|\theta\|_2^2$$

Example: online linear regression

BUT, a very intuitive algorithm actually achieves no-regret property:

Every iteration t :

1. Learner first picks θ_t that minimizes the aggregated loss

$$\theta_t = \arg\min_{\theta \in \text{Ball}} \sum_{i=0}^{t-1} \left(\theta^\top x_i - y_i\right)^2 + \lambda \|\theta\|_2^2$$

This is called Follow-the-Regularized-Leader (FTRL), and it achieves no-regret property:

Example: online linear regression

BUT, a very intuitive algorithm actually achieves no-regret property:

Every iteration t :

1. Learner first picks θ_t that minimizes the aggregated loss

$$\theta_t = \arg\min_{\theta \in \text{Ball}} \sum_{i=0}^{t-1} \left(\theta^\top x_i - y_i\right)^2 + \lambda \|\theta\|_2^2$$

This is called Follow-the-Regularized-Leader (FTRL), and it achieves no-regret property:

$$\sum_{i=0}^{T-1} \ell_i(\theta_i) - \min_{\theta \in \text{Ball}} \sum_{i=0}^{T-1} \ell_i(\theta) = O\left(1/\sqrt{T}\right)$$

Generally, Follow-the-Regularized-Leader is no-regret

At time step t, learner has seen $\ell_0, \ldots \ell_{t-1}$, which new decision she could pick?

$$\textbf{FTL: } \theta_t = \min_{\theta \in \Theta} \sum_{i=0}^{t-1} \ell_i(\theta) + \lambda R(\theta)$$

Generally, Follow-the-Regularized-Leader is no-regret

At time step t, learner has seen $\ell_0, \ldots \ell_{t-1}$, which new decision she could pick?

$$\textbf{FTL: } \theta_t = \min_{\theta \in \Theta} \sum_{i=0}^{t-1} \ell_i(\theta) + \lambda R(\theta)$$

Theorem (FTL) (optional): if Θ is convex, and ℓ_t is convex for all t, and $R(\theta)$ is strongly convex, then for regret of FTL, we have:

$$\frac{1}{T}\left[\sum_{t=0}^{T-1} \ell_t(\theta_t) - \min_{\theta \in \Theta} \sum_{t=0}^{T-1} \ell_t(\theta)\right] = O\left(1/\sqrt{T}\right)$$

Any questions about no-regret online learning?

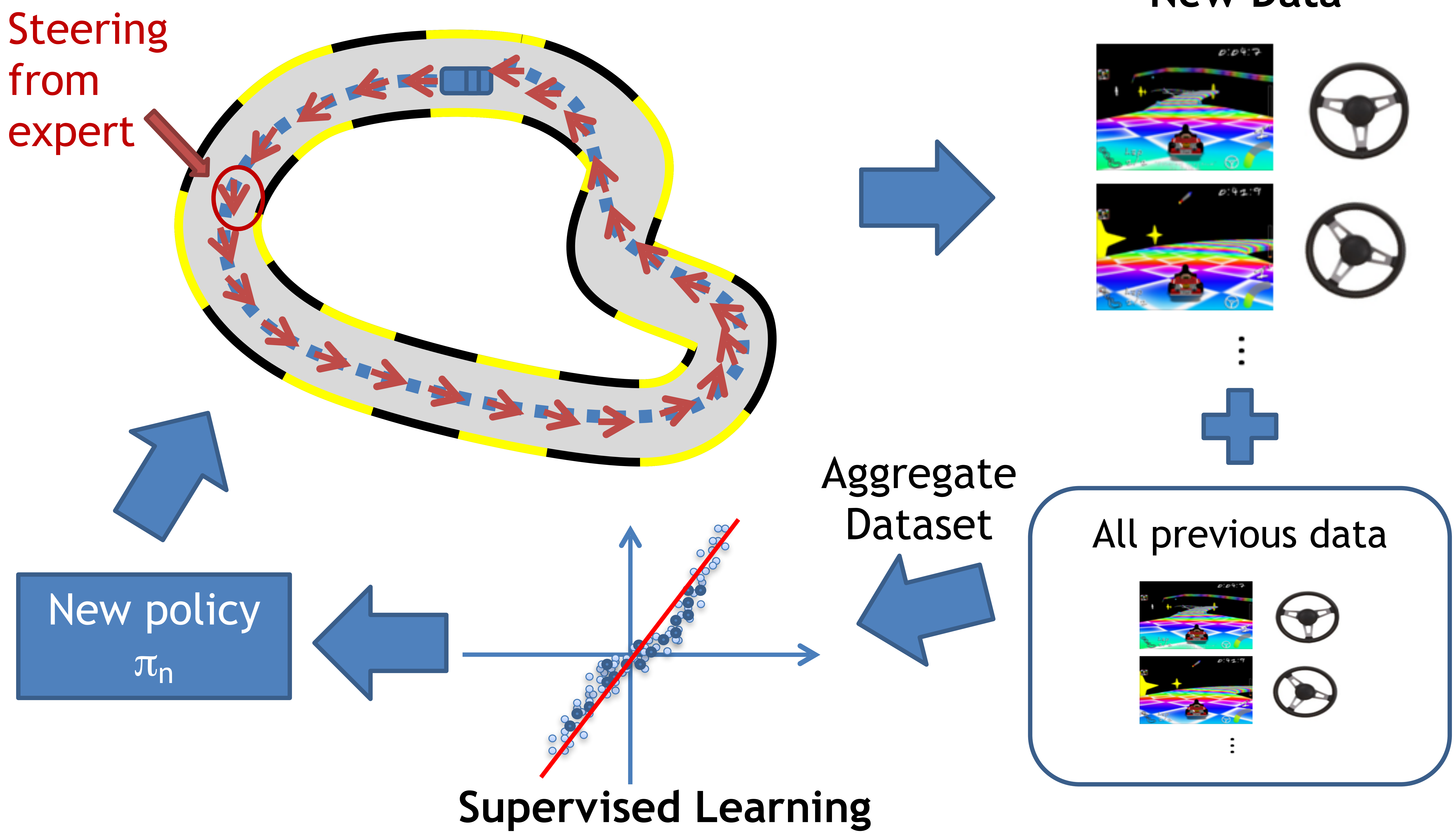
Online learning is a very rich research area — details are out of scope

Key message:

Learner has to make a decision before Adversary picks a loss function,
yet it is possible to do as well as the best decision in hindsight if we had access to all the loss functions beforehand

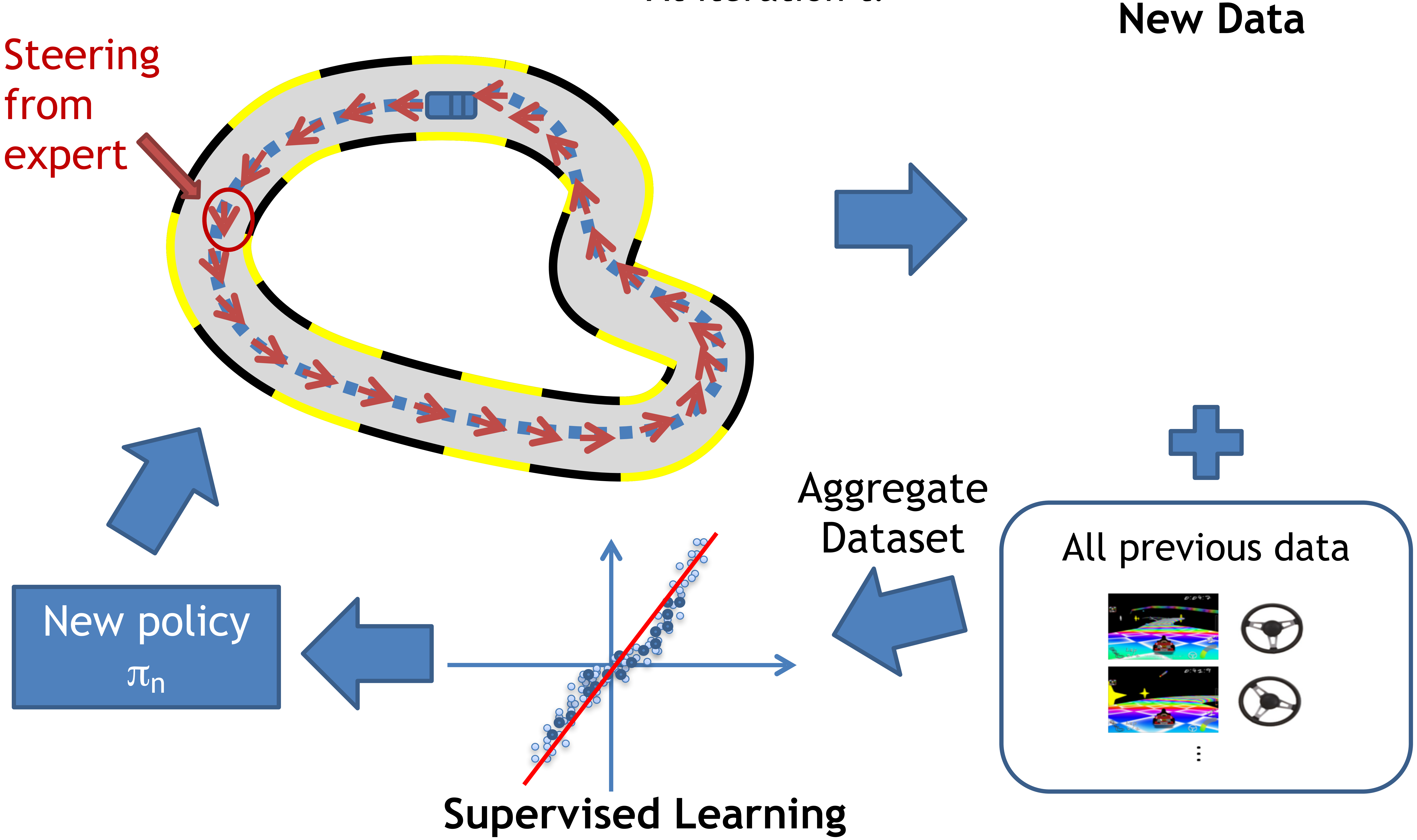
DAgger Revisit

At iteration t:



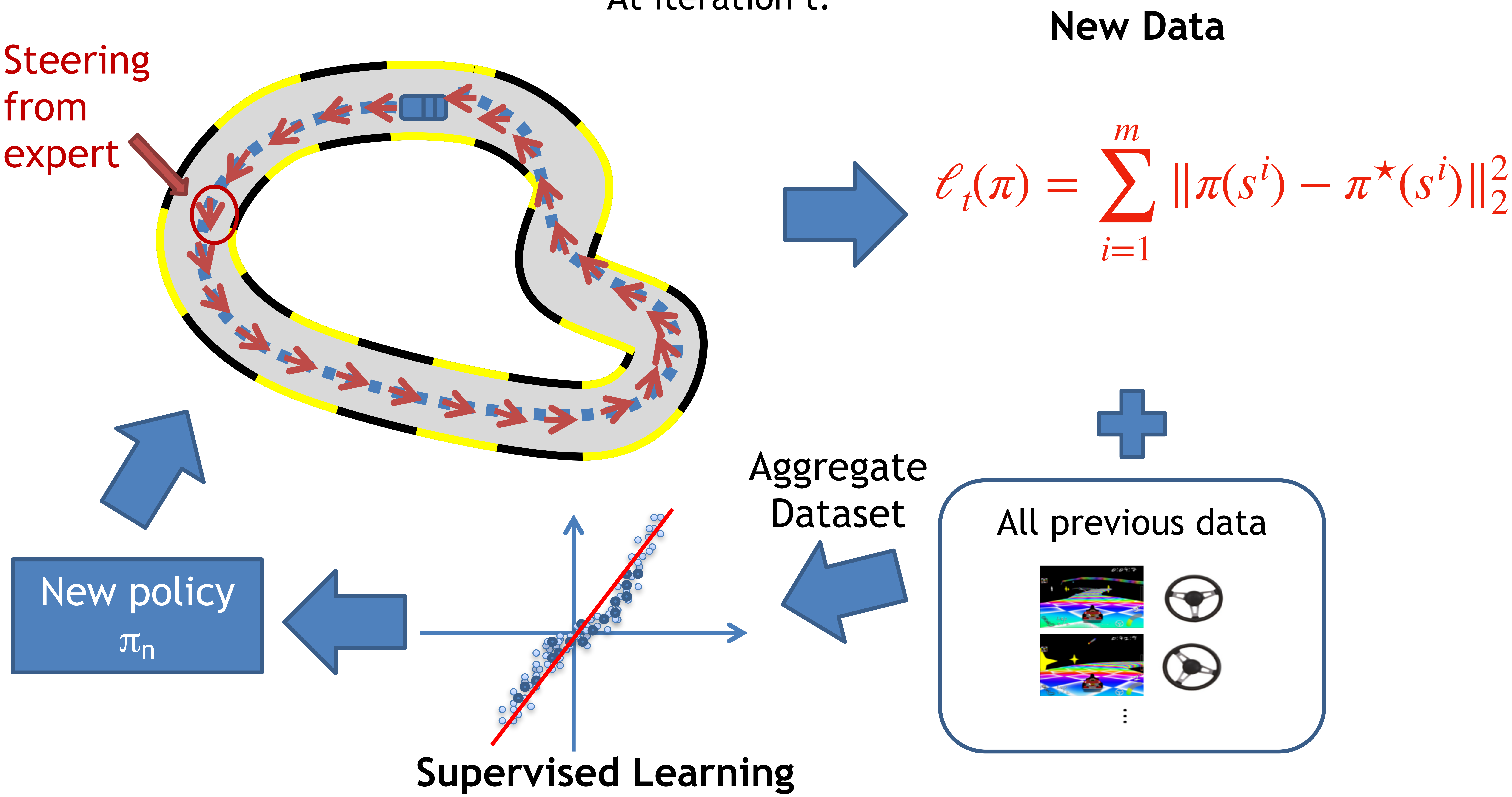
DAgger Revisit

At iteration t:



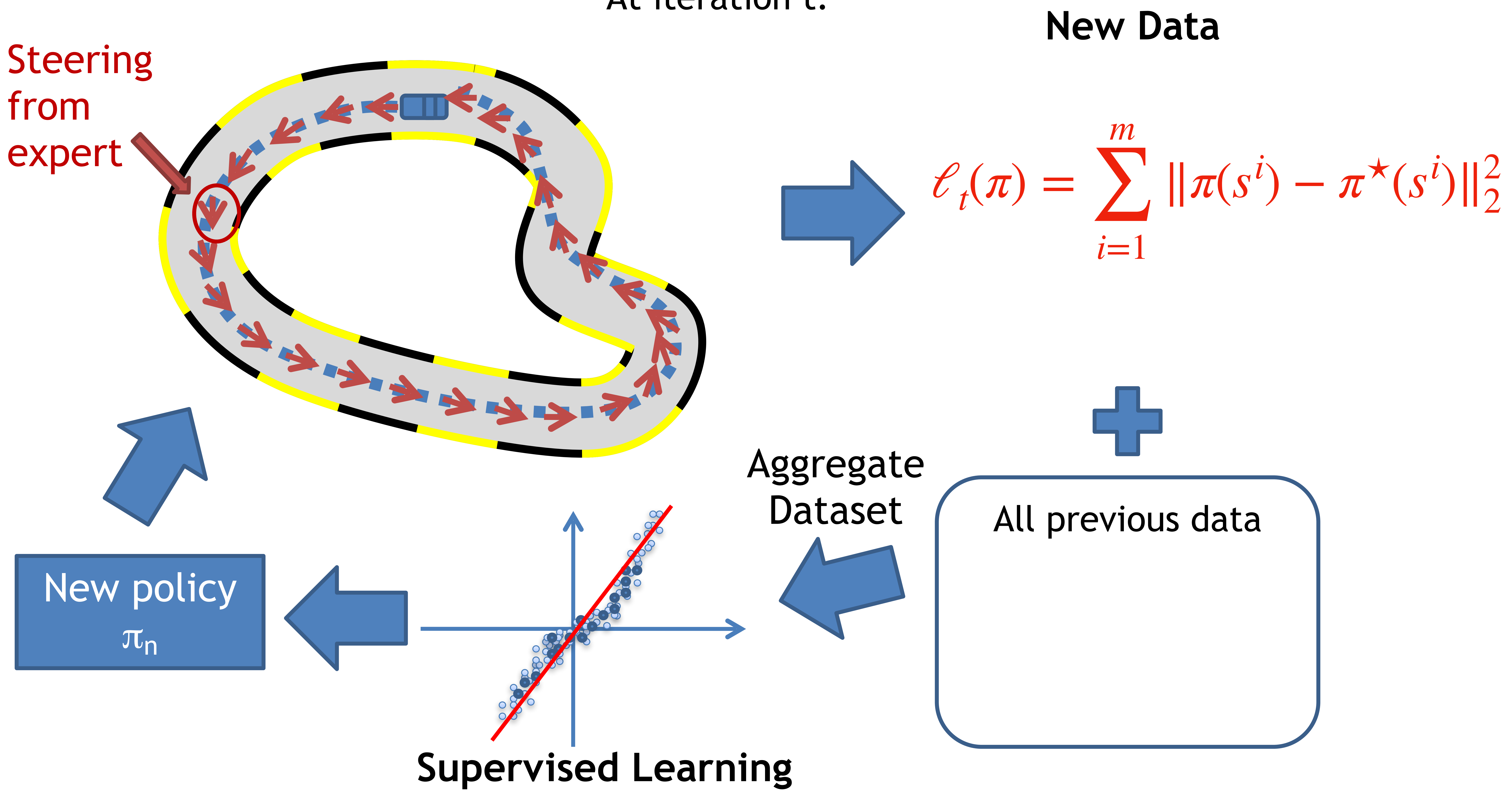
DAgger Revisit

At iteration t:



DAgger Revisit

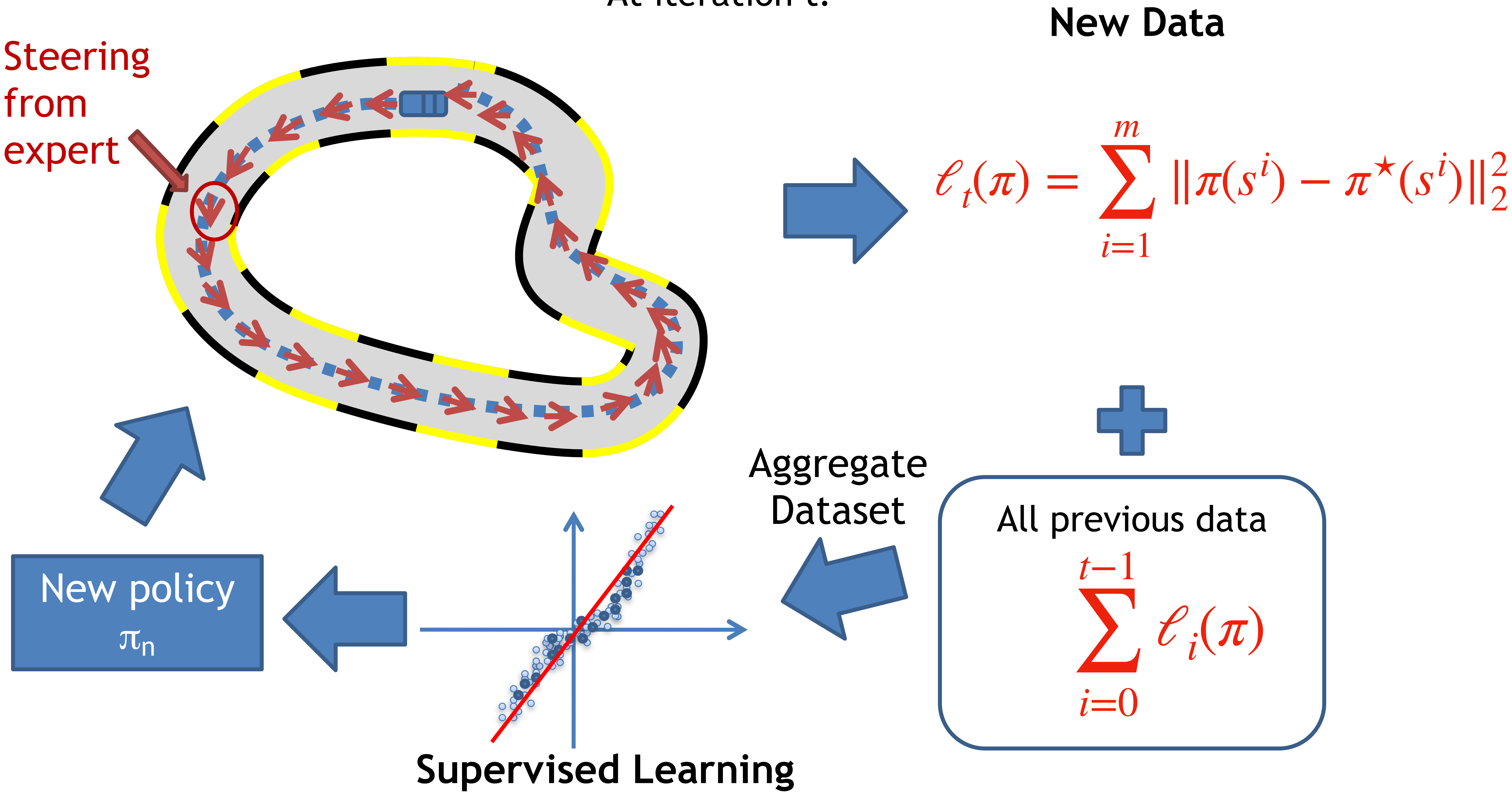
At iteration t:



$$\ell_t(\pi) = \sum_{i=1}^{m} \|\pi(s^i) - \pi^\star(s^i)\|_2^2$$

DAgger Revisit

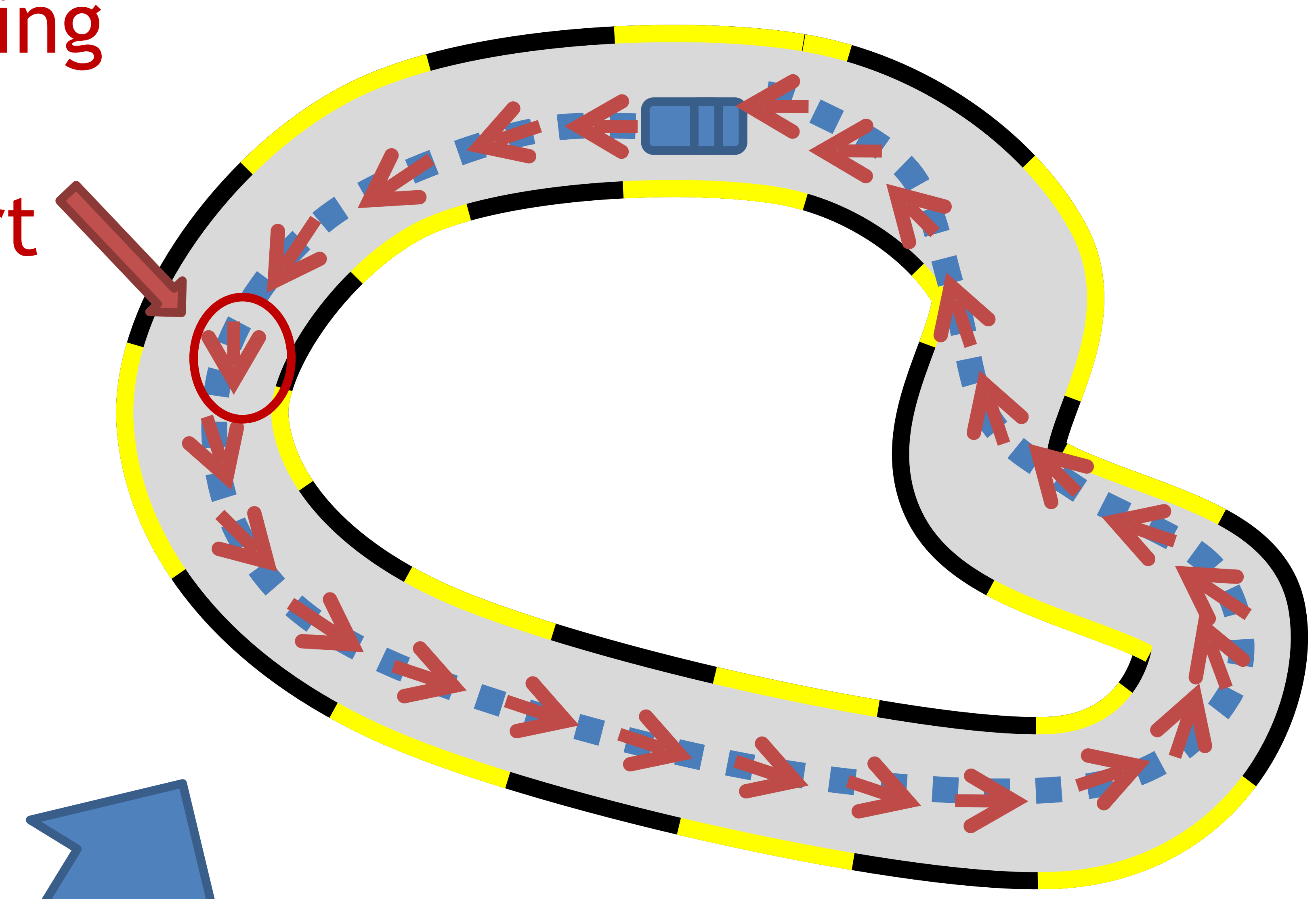
At iteration t:



DAgger Revisit

At iteration t:

Steering from expert



New Data

$$\ell_t(\pi) = \sum_{i=1}^{m} \|\pi(s^i) - \pi^\star(s^i)\|_2^2$$

+

All previous data

$$\sum_{i=0}^{t-1} \ell_i(\pi)$$

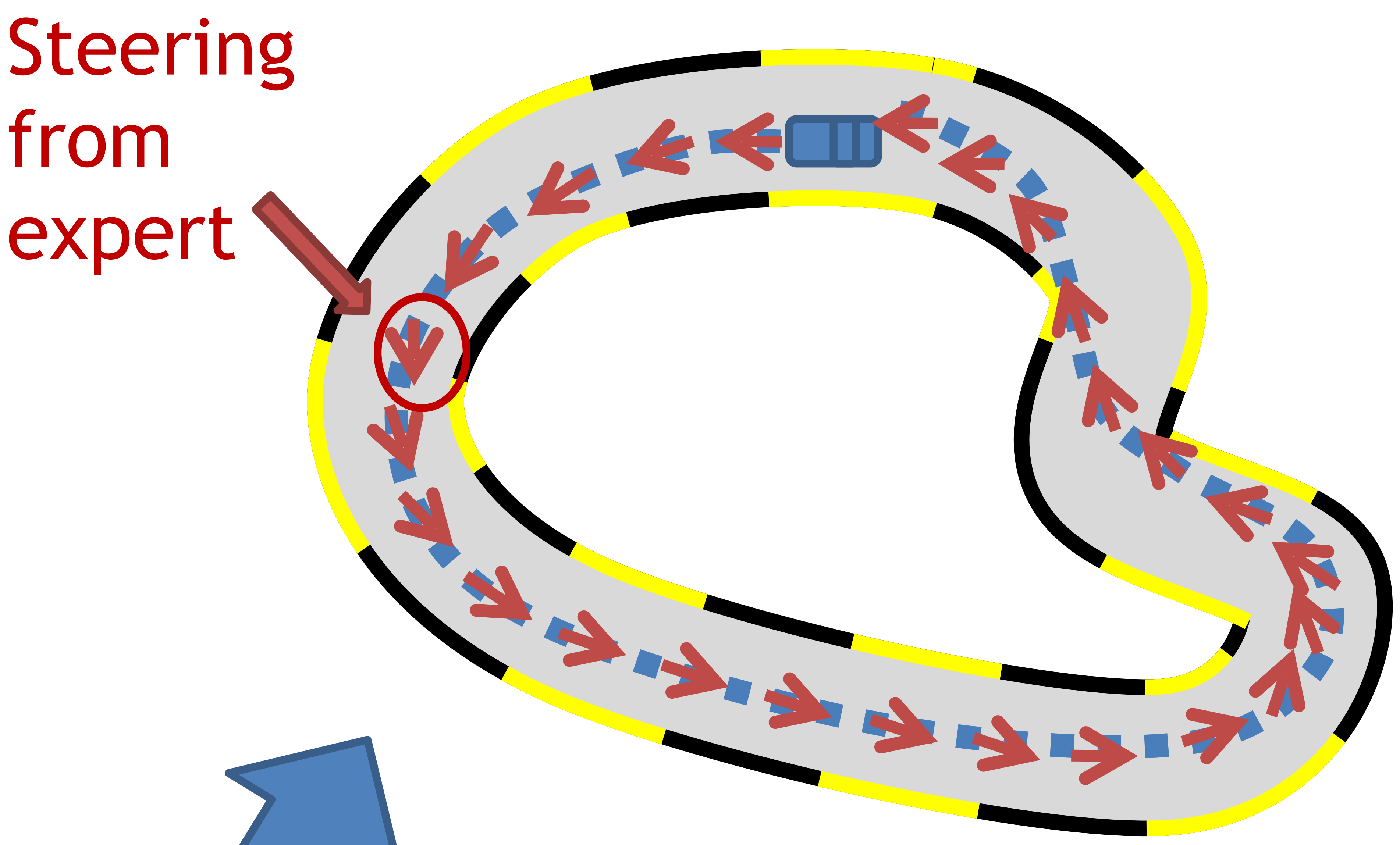
Aggregate Dataset

New policy π_n

Supervised Learning

DAgger Revisit

At iteration t:



New Data

$$\ell_t(\pi) = \sum_{i=1}^{m} \|\pi(s^i) - \pi^\star(s^i)\|_2^2$$

All previous data

$$\sum_{i=0}^{t-1} \ell_i(\pi)$$

Aggregate Dataset

$$\pi_t = \arg\min_{\pi} \sum_{i=0}^{t-1} \ell_i(\pi) + \lambda R(\pi)$$

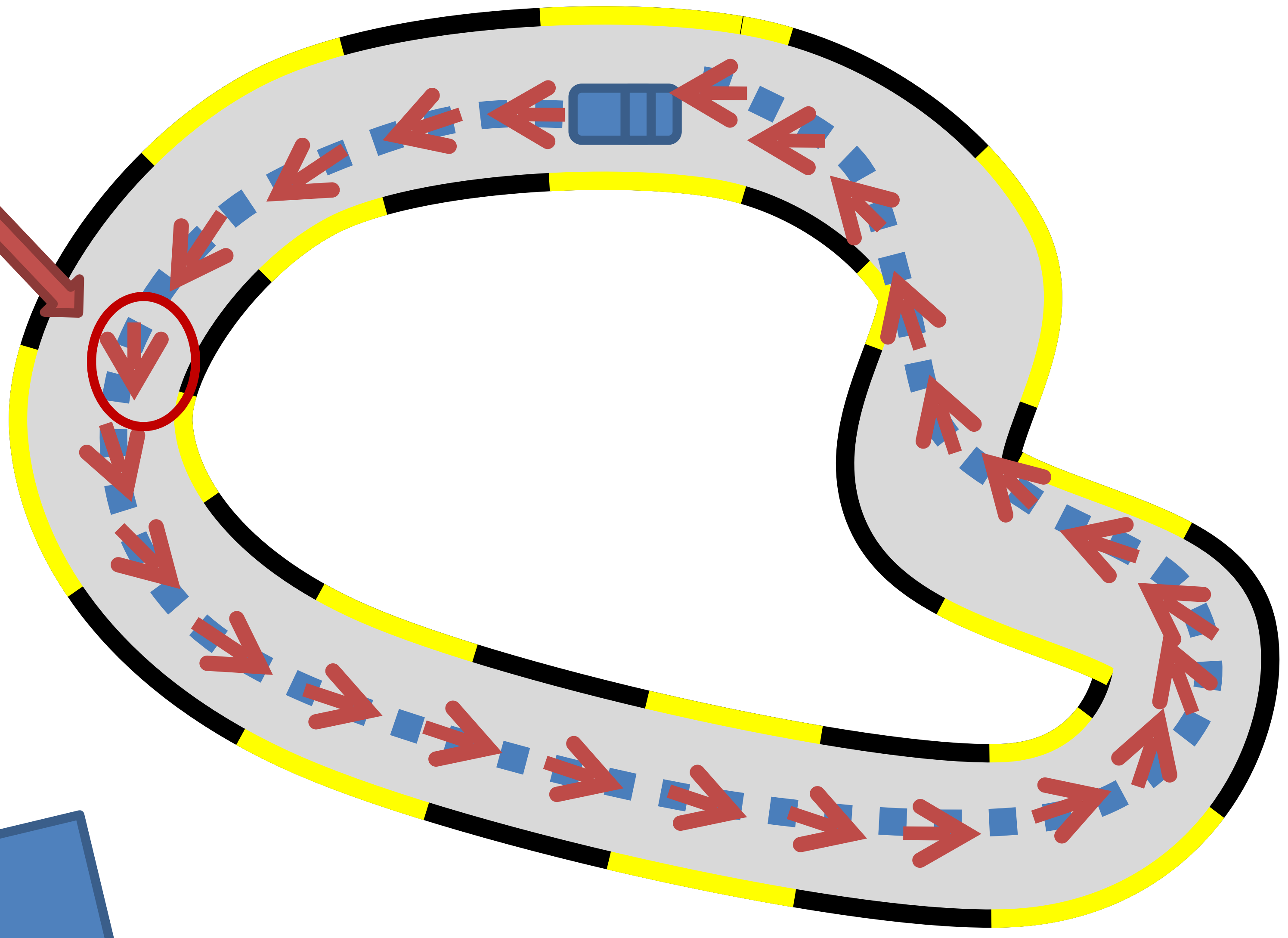
Supervised Learning

New policy π_n

DAgger Revisit

At iteration t:

Steering from expert



New Data

$$\ell_t(\pi) = \sum_{i=1}^{m} \|\pi(s^i) - \pi^\star(s^i)\|_2^2$$

+

All previous data

$$\sum_{i=0}^{t-1} \ell_i(\pi)$$

Aggregate Dataset

$$\pi_t = \arg\min_{\pi} \sum_{i=0}^{t-1} \ell_i(\pi) + \lambda R(\pi)$$

New policy π_n

Supervised Learning

Data Aggregation = Follow-the-Regularized-Leader Online Learner

Summary for Today

1. The DAgger algorithm

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

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

1. W/ π^t, generate dataset $\mathscr{D}^t = \{s_i, a_i^\star\}, s_i \sim d_\mu^{\pi^t}, a_i^\star = \pi^\star(s_i)$
2. **Data aggregation**: $\mathscr{D} = \mathscr{D} + \mathscr{D}^t$
3. **Update policy via Supervised-Learning**: $\pi^{t+1} = \text{SL}\left(\mathscr{D}\right)$

Summary for Today

1. The DAgger algorithm

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

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

1. W/ π^t, generate dataset $\mathcal{D}^t = \{s_i, a_i^\star\}, s_i \sim d_\mu^{\pi^t}, a_i^\star = \pi^\star(s_i)$
2. **Data aggregation**: $\mathcal{D} = \mathcal{D} + \mathcal{D}^t$
3. **Update policy via Supervised-Learning**: $\pi^{t+1} = \text{SL}\left(\mathcal{D}\right)$

2. We can see that DAgger is essentially an online-learning algorithm (FTRL)