Imitation Learning: Behavior Cloning, Distribution Shift, & Distribution Matching

Sham Kakade and Wen Sun

CS 6789: Foundations of Reinforcement Learning

Today: Imitation Learning

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The hybrid Setting: Statistical Benefit and Distribution Matching

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

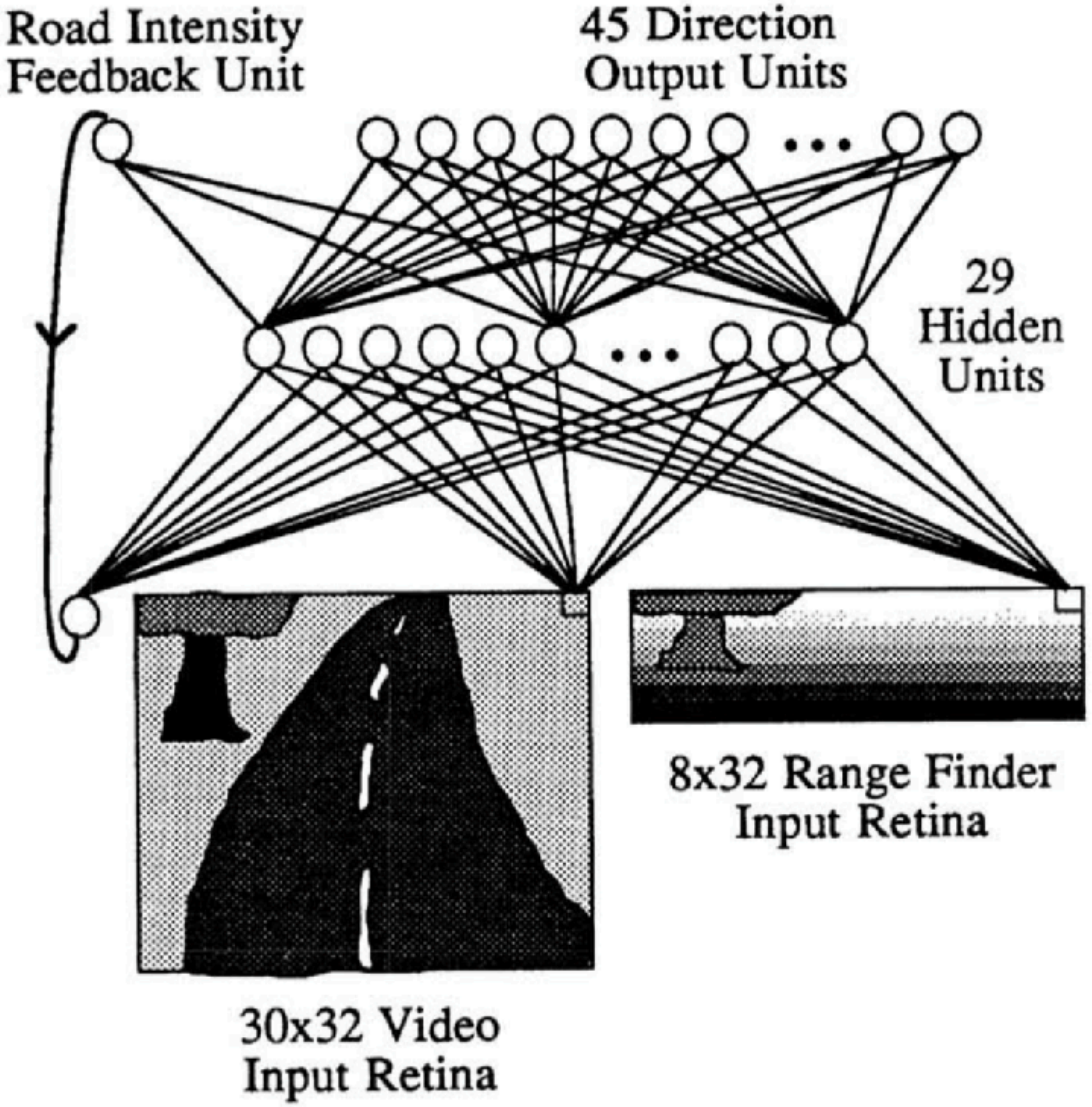
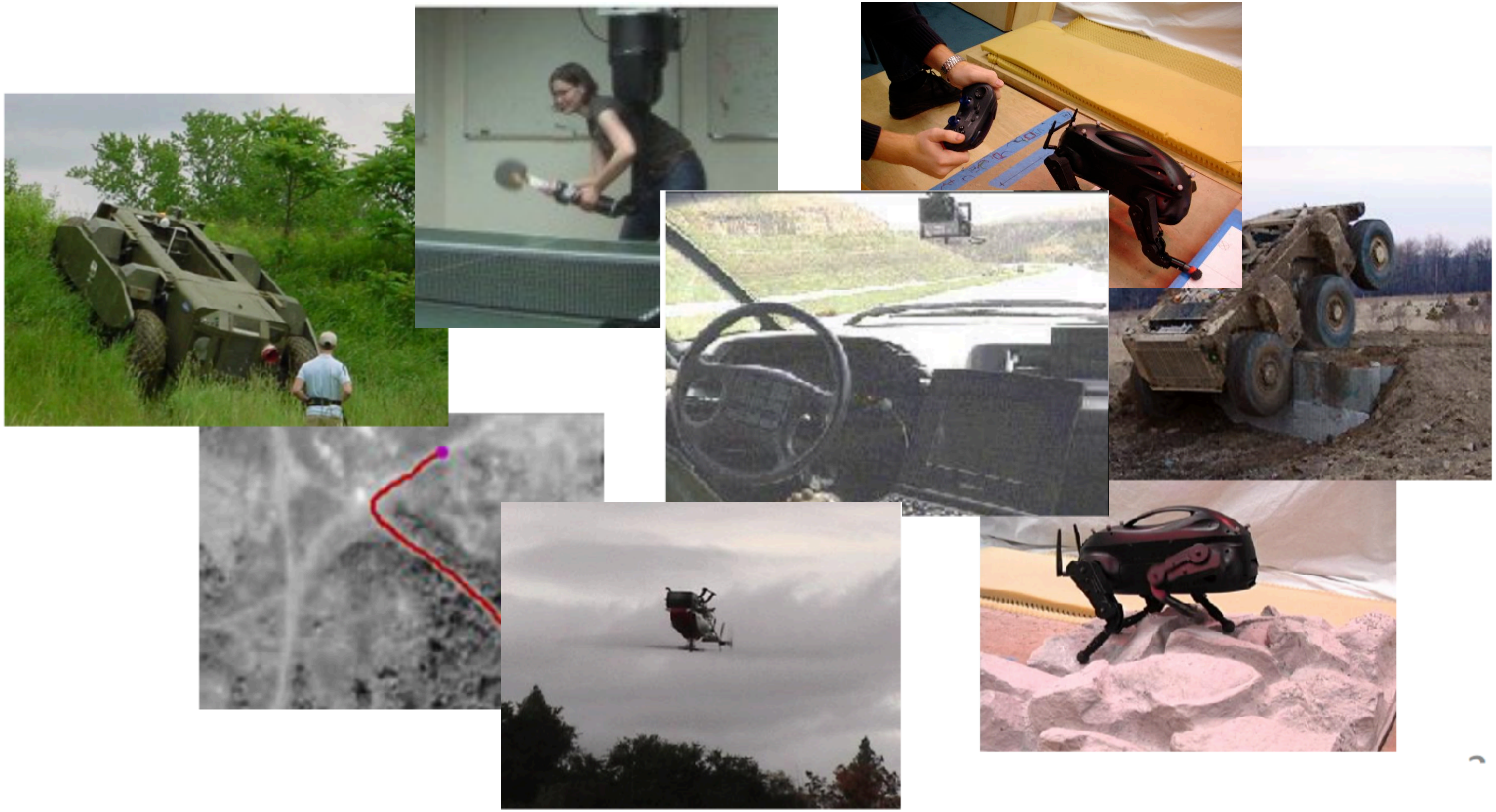


Figure 1: ALVINN Architecture

Imitation Learning



Imitation Learning

Imitation Learning

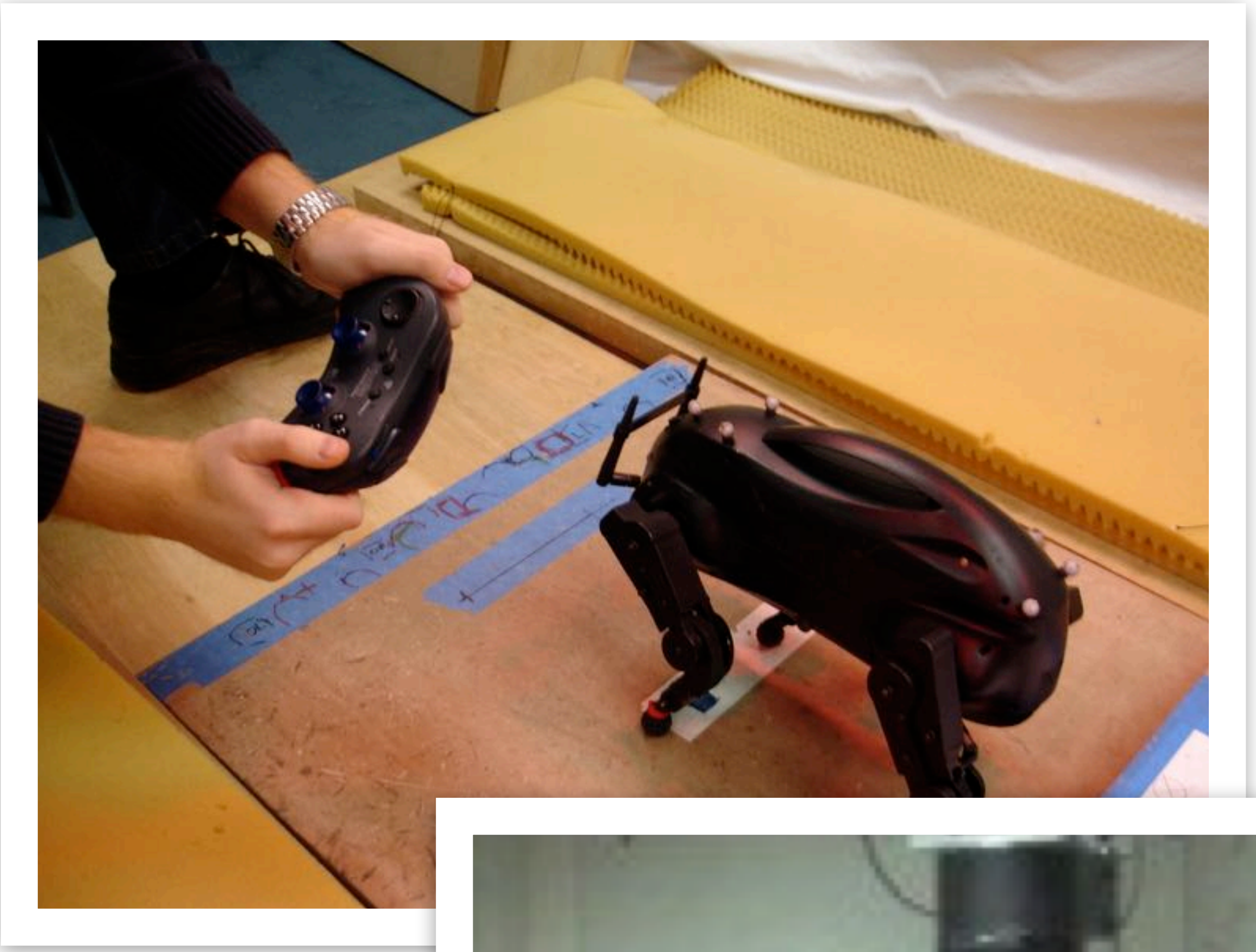
Imitation Learning

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:

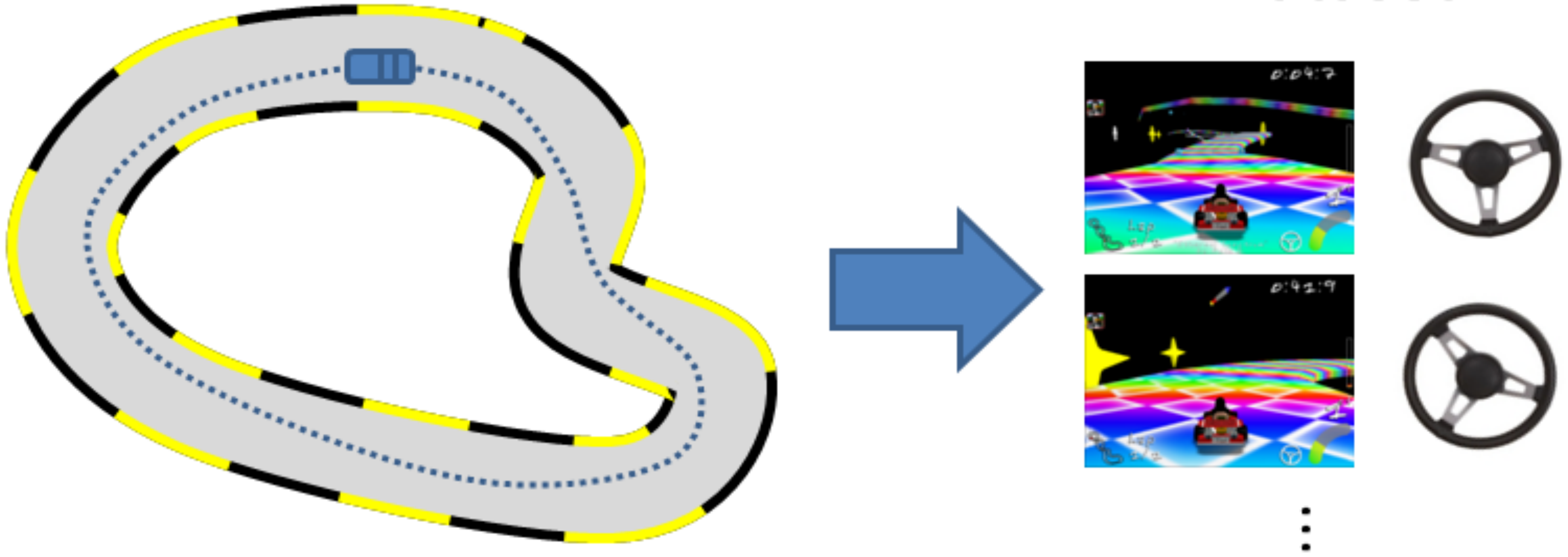
Steering Angle
in [-1, 1]

[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning

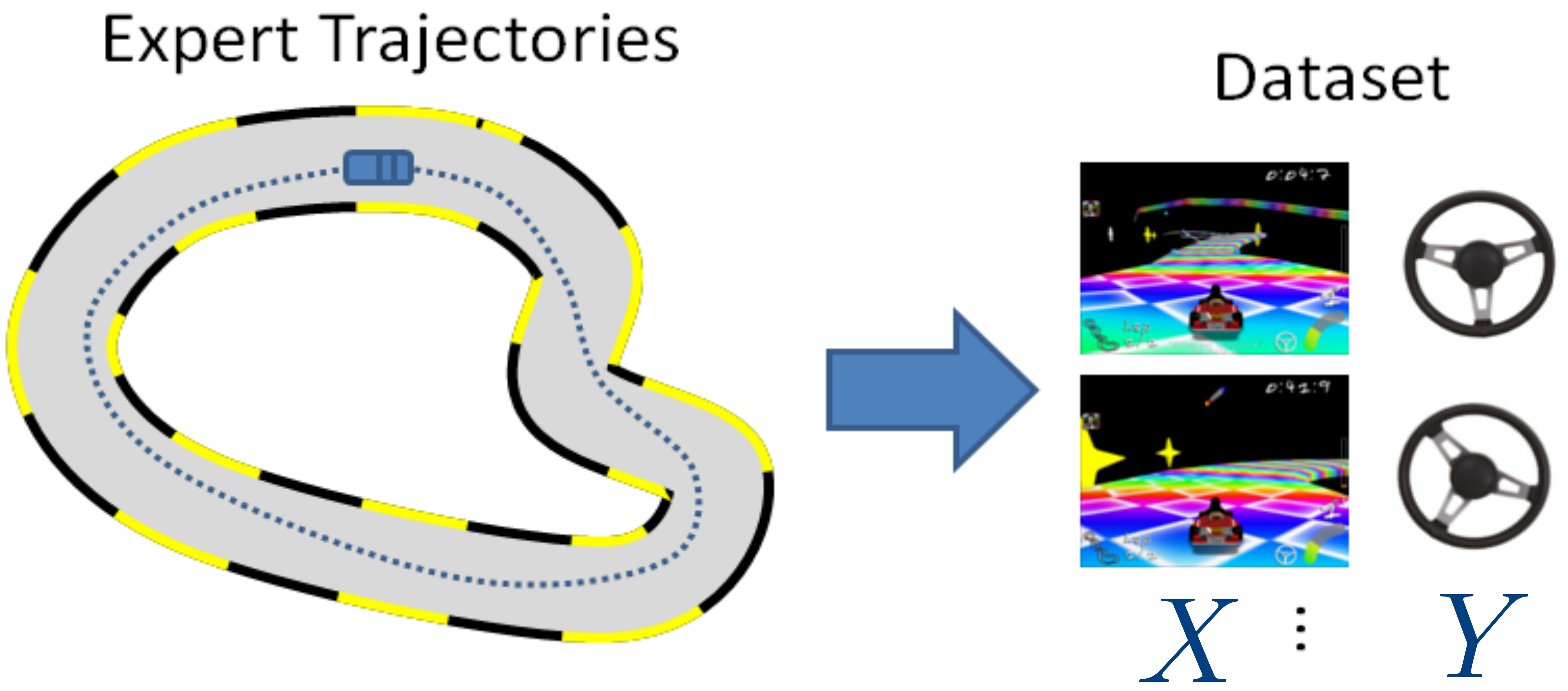
Expert Trajectories

Dataset



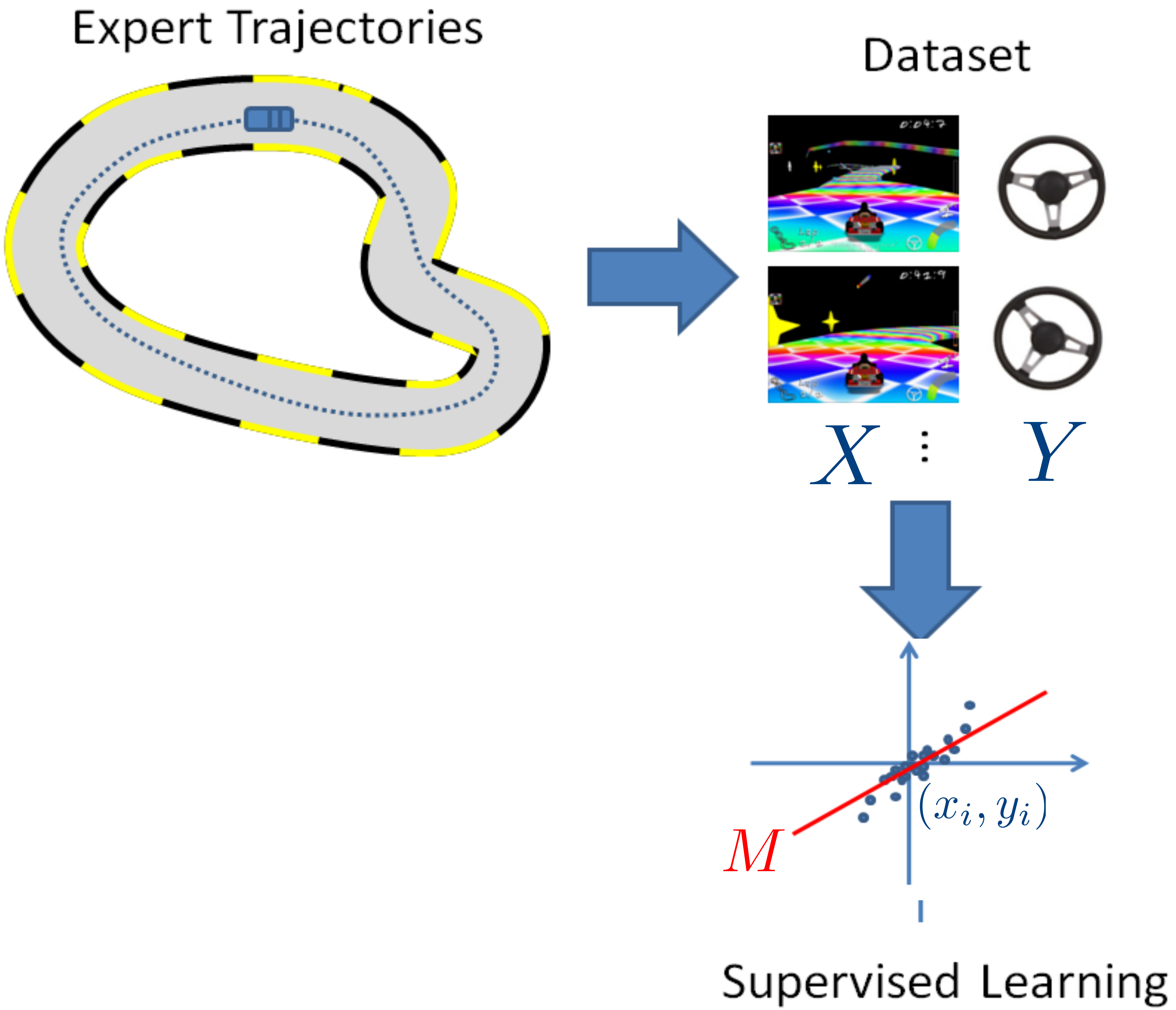
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



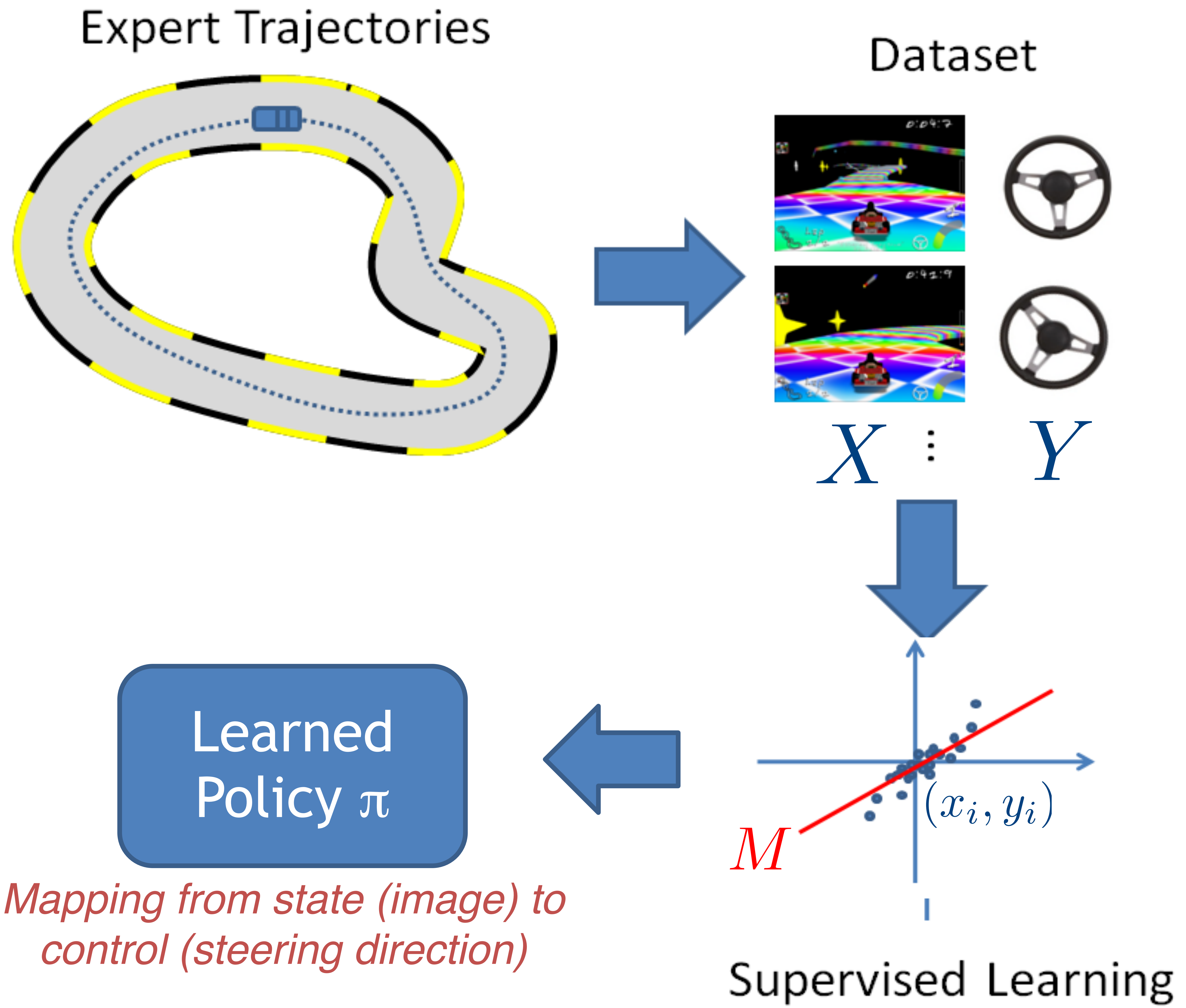
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning




Ready!


Ready!


Ready!

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

Ground truth reward $r(s, a) \in [0,1]$ is unknown; assume expert is a near 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; assume expert is a near 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; assume expert is a near 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 with Maximum Likelihood Estimation (MLE):

Let’s formalize the Behavior Cloning algorithm

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

BC with Maximum Likelihood Estimation (MLE):

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Let's formalize the Behavior Cloning algorithm

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

BC with Maximum Likelihood Estimation (MLE):

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

(We can reduce it to other supervised learning oracles such as classification, regression)

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Analysis

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Analysis

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Analysis

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

Theorem [MLE Guarantee] With probability at least $1 - \delta$, we have:

$$\mathbb{E}_{s \sim d^{\pi^\star}} \left\| \widehat{\pi}(\cdot \mid s) - \pi^\star(\cdot \mid s) \right\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Analysis

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

Theorem [MLE Guarantee] With probability at least $1 - \delta$, we have:

$$\mathbb{E}_{s \sim d^{\pi^\star}} \left\| \widehat{\pi}(\cdot \mid s) - \pi^\star(\cdot \mid s) \right\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

This $1/\sqrt{M}$ rate should be expected:
no training and testing mismatch at this stage!

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Analysis

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 2: Transfer supervised learning error to policy's performance gap

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 2: Transfer supervised learning error to policy's performance gap

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

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

Analysis

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

Assumption: Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 2: Transfer supervised learning error to policy's performance gap

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

$$V^{\pi^\star} - V^{\hat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

Note that $1/(1-\gamma)^2$ quadratic dependency on effective horizon

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

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

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

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

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

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

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

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

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

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

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

Analysis

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

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

$$V^{\pi^\star} - V^{\hat{\pi}} \le \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

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

$$= \mathbb{E}_{s\sim d^{\pi^\star}} \mathbb{E}_{a\sim\pi^\star(\cdot|s)} A^{\hat{\pi}}(s,a) - \mathbb{E}_{s\sim d^{\pi^\star}} \mathbb{E}_{a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a)$$

$$\le \mathbb{E}_{s\sim d^{\pi^\star}} \frac{1}{1-\gamma} \left\| \pi^\star(\cdot \mid s) - \hat{\pi}(\cdot \mid s) \right\|_1$$

Analysis

$$\widehat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^\star \mid s_i^\star)$$

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

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \underbrace{\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}}_{\text{MLE error}}$$

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

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

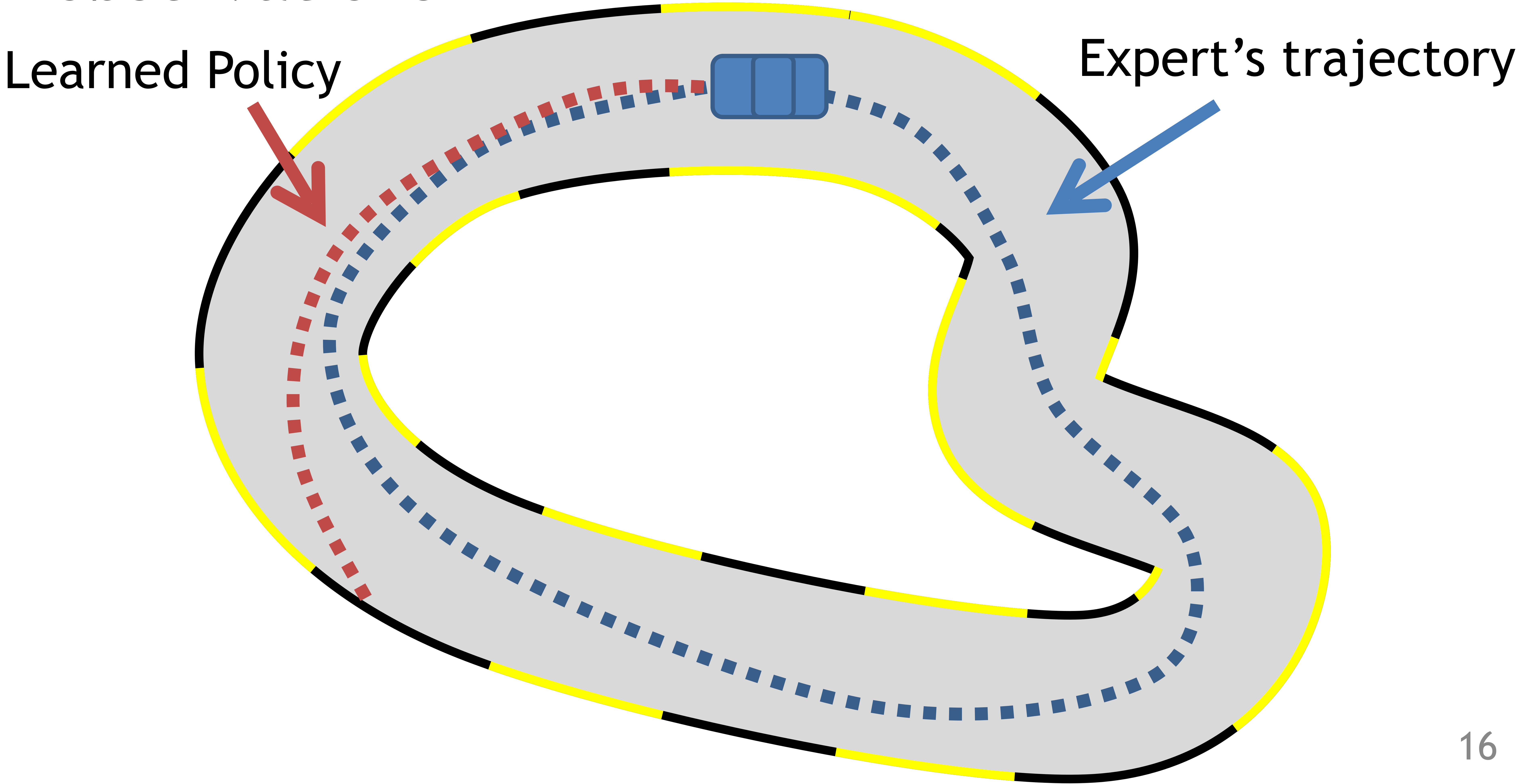
$$\leq \mathbb{E}_{s\sim d^{\pi^\star}} \frac{1}{1-\gamma} \left\| \pi^\star(\cdot \mid s) - \widehat{\pi}(\cdot \mid s) \right\|_1$$

$$\leq \frac{2}{1-\gamma} \mathbb{E}_{s\sim d^{\pi^\star}} \|\pi^\star(\cdot \mid s) - \widehat{\pi}(\cdot \mid s)\|_{tv}$$

What could go wrong?

[Pomerleau89,Daume09]

- Predictions affect future inputs/ observations



Distribution Shift: Intuitive Explanation

Let's just focus on finite horizon (H) and deterministic policies here:

$$\mathbb{E}_{s\sim d_h^{\pi^\star}} \widehat{\pi}(s) \neq \pi^\star(s) \leq \epsilon, \forall h$$

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

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"

A potential Fix

A potential Fix

Let’s roll out our policy in the real world, and compare our trajectories to the expert’s trajectories, and then refine our learned model.

The Hybrid Imitation Learning Setting:

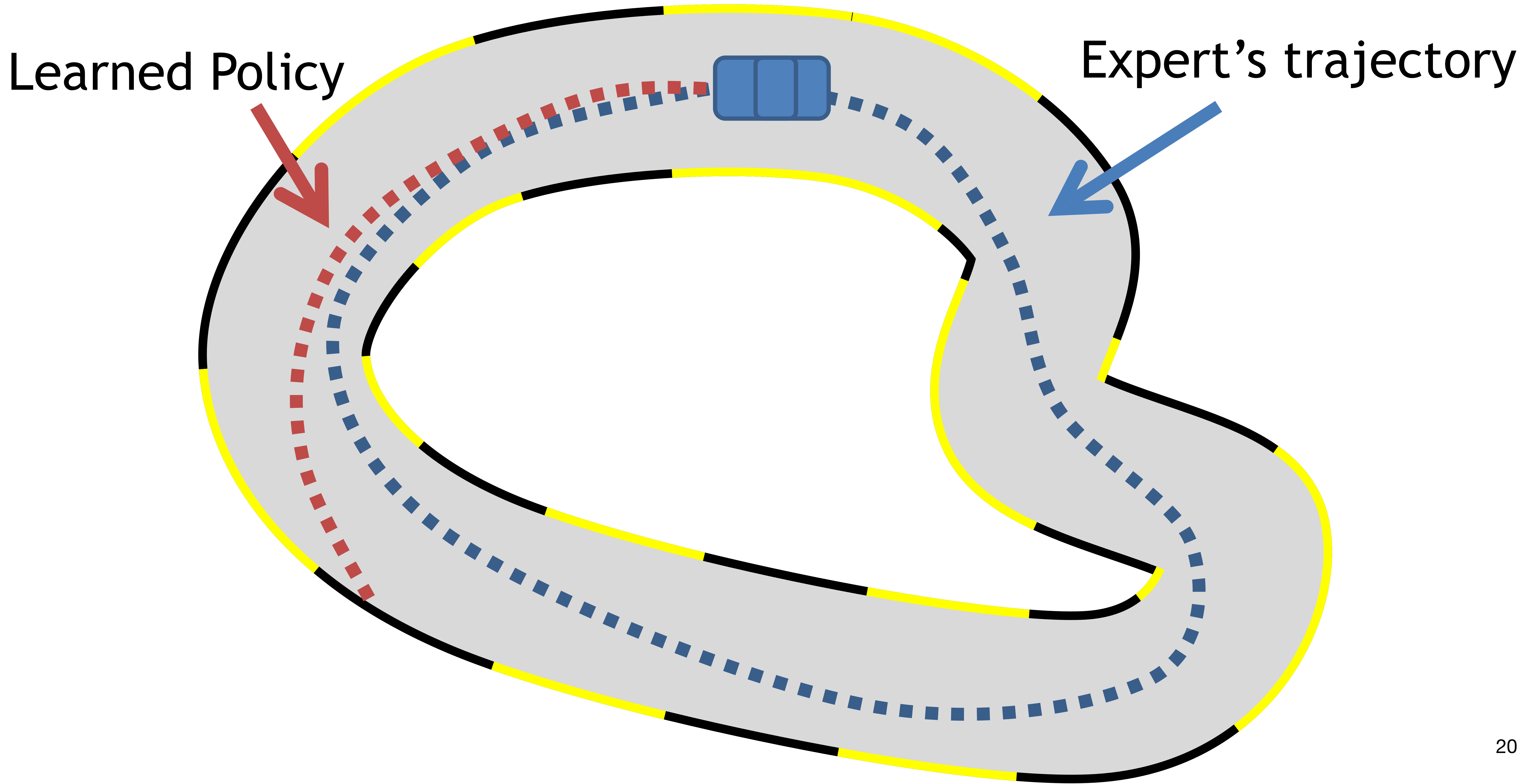
Recall BC: we only use offline expert data—no interaction with the environment

Hybrid setting: offline expert data + simulator (e.g., known transition P)

The Hybrid Imitation Learning Setting:

Recall BC: we only use offline expert data—no interaction with the environment

Hybrid setting: offline expert data + simulator (e.g., known transition P)



Let's formalize the Hybrid Setting

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; assume expert is the 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 Hybrid Setting

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; assume expert is the optimal policy $\pi^\star$

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

This time, we have a known transition P (but we cannot plan because r is unknown)

Let's formalize the Hybrid Setting

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; assume expert is the optimal policy $\pi^\star$

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

This time, we have a known transition P (but we cannot plan because r is unknown)

Key Q: can we do better than offline IL Behavior Cloning (statistically at least—assuming infinite computation power)?

Key Idea: distribution matching

Integral probability metric (IPM)

Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathscr{F} = \{f : X \mapsto \mathbb{R}\}$, and two distributions p_1 and p_2

Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathscr{F} = \{f : X \mapsto \mathbb{R}\}$, and two distributions p_1 and p_2

$$\text{IPM}_{\mathscr{F}}(p_1, p_2) = \max_{f \in \mathscr{F}} \left[\mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x) \right]$$

Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathcal{F} = \{f : X \mapsto \mathbb{R}\}$, and two distributions p_1 and p_2

$$\text{IPM}_{\mathcal{F}}(p_1, p_2) = \max_{f \in \mathcal{F}} \left[\mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x) \right]$$

$$\mathcal{F} = \{f : \|f\|_\infty \leq 1\} \Rightarrow \text{IPM}_{\mathcal{F}}(p_1, p_2) := \|p_1 - p_2\|_{tv}$$

Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathscr{F} = \{f : X \mapsto \mathbb{R}\}$, and two distributions p_1 and p_2

$$\text{IPM}_{\mathscr{F}}(p_1, p_2) = \max_{f \in \mathscr{F}} \left[\mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x) \right]$$

$$\mathscr{F} = \{f : \|f\|_\infty \leq 1\} \Rightarrow \text{IPM}_{\mathscr{F}}(p_1, p_2) := \|p_1 - p_2\|_{tv}$$

$$\mathscr{F} = \{f : f \text{ is 1-Lipschitz}\} \Rightarrow \text{IPM}_{\mathscr{F}}(p_1, p_2) := \text{wasserstein dis}(p_1, p_2)$$

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step 2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathscr{F}} \left[\mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step 2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathscr{F}} \left[\mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\widetilde{\mathscr{F}} := \{f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi'\}$

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step 2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathscr{F}} \left[\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi'}} f(s,a)\right]$$

Q: what is the size of $\widetilde{\mathscr{F}}$?

Set refined discriminator class $\widetilde{\mathscr{F}} := \{f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi'\}$

Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
Π is discrete, and realizable, i.e., $\pi^\star \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step 2: select useful discriminators: for all pair $\pi \,\&\, \pi'$, with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathscr{F}} \left[\mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Q: what is the size of $\widetilde{\mathscr{F}}$?

Set refined discriminator class $\widetilde{\mathscr{F}} := \{f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi'\}$

$$\forall \pi \,\&\, \pi' \in \Pi,\ \|d^\pi - d^{\pi'}\|_{tv} = \max_{f \in \widetilde{\mathscr{F}}} \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a)$$

Algorithm: Distribution Matching TV distance

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\pi}} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\widetilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi' \}$

Algorithm: Distribution Matching TV distance

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathcal{F}} \left[\mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\widetilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi' \}$

Step 3: Select a policy using expert dataset $\mathcal{D} = \{ s_i^\star, a_i^\star \}_{i=1}^M$

Algorithm: Distribution Matching TV distance

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall P is known);
(This step is computationally inefficient)

Step2: select useful discriminators: for all pair π & π', with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg\max_{f \in \mathcal{F}} \left[\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\widetilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \,\&\, \pi' \in \Pi, \pi \neq \pi' \}$

Step 3: Select a policy using expert dataset $\mathcal{D} = \{s_i^\star, a_i^\star\}_{i=1}^M$

$$\widehat{\pi} := \arg\min_{\pi \in \Pi} \left[\max_{f \in \widetilde{\mathcal{F}}} \left[\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \frac{1}{M} \sum_{i=1}^M f(s_i^\star, a_i^\star) \right] \right]$$

Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1-\delta$, our algorithm finds a policy $\widehat{\pi}$, s.t.,

$$V^{\pi^\star} - V^{\widehat{\pi}} \le \mathcal{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}\right)$$

Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1 - \delta$, our algorithm finds a policy $\widehat{\pi}$, s.t.,

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \mathcal{O}\left(\frac{1}{1-\gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$

1. Key step is to prove: $\left\| d^{\widehat{\pi}} - d^{\pi^\star} \right\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$

Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1 - \delta$, our algorithm finds a policy $\widehat{\pi}$, s.t.,

$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \mathcal{O}\left(\frac{1}{1-\gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$

1. Key step is to prove: $\left\| d^{\widehat{\pi}} - d^{\pi^\star} \right\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$

2. For performance: $V^{\pi^\star} - V^{\pi} \leq \frac{1}{1-\gamma} \left[\mathbb{E}_{s,a\sim d^{\pi^\star}} r(s,a) - \mathbb{E}_{s,a\sim d^{\widehat{\pi}}} r(s,a) \right]$

Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1-\delta$, our algorithm finds a policy $\widehat{\pi}$, s.t.,

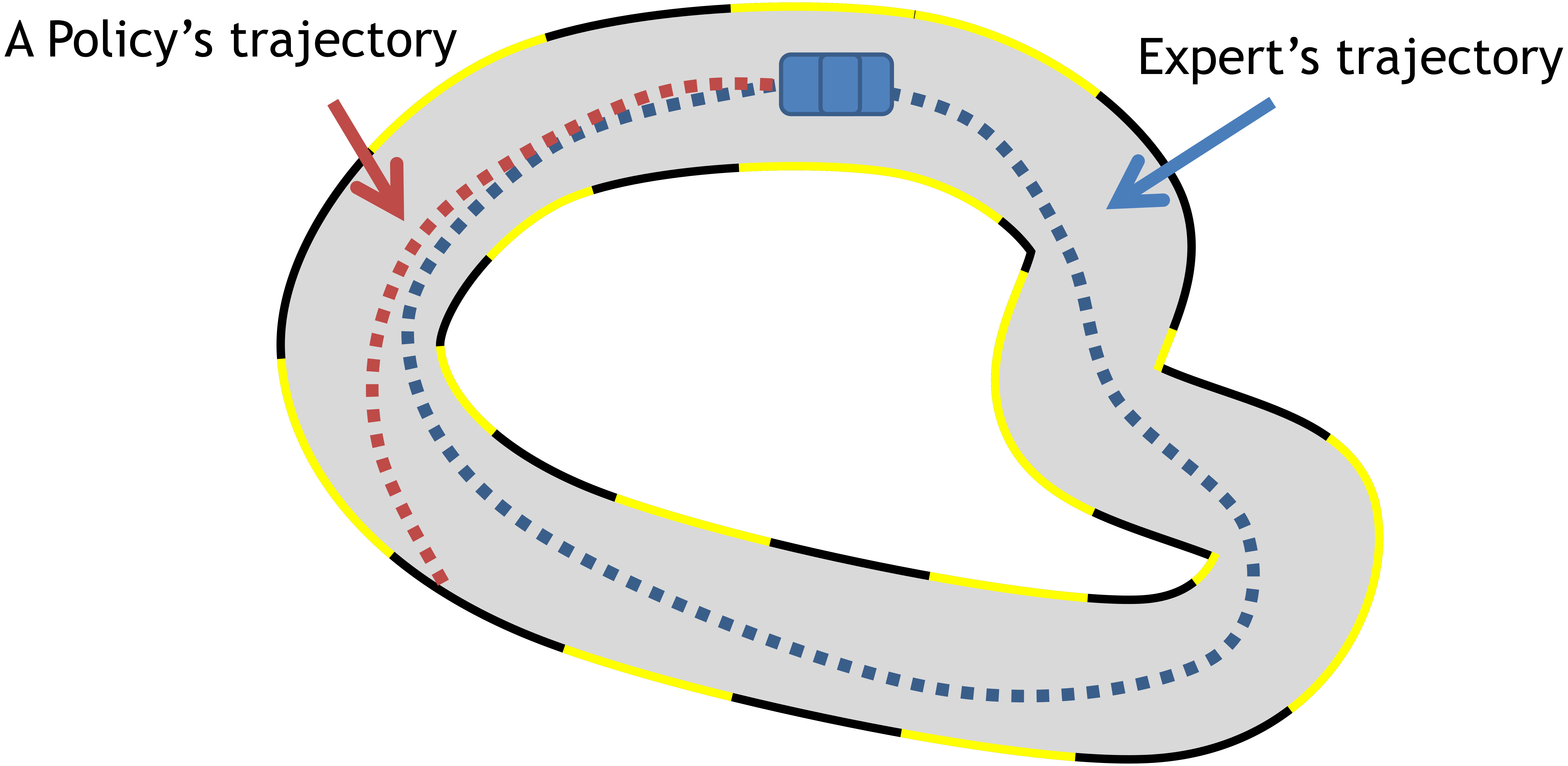
$$V^{\pi^\star} - V^{\widehat{\pi}} \leq \mathcal{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}\right)$$

1. Key step is to prove: $\left\| d^{\widehat{\pi}} - d^{\pi^\star} \right\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$

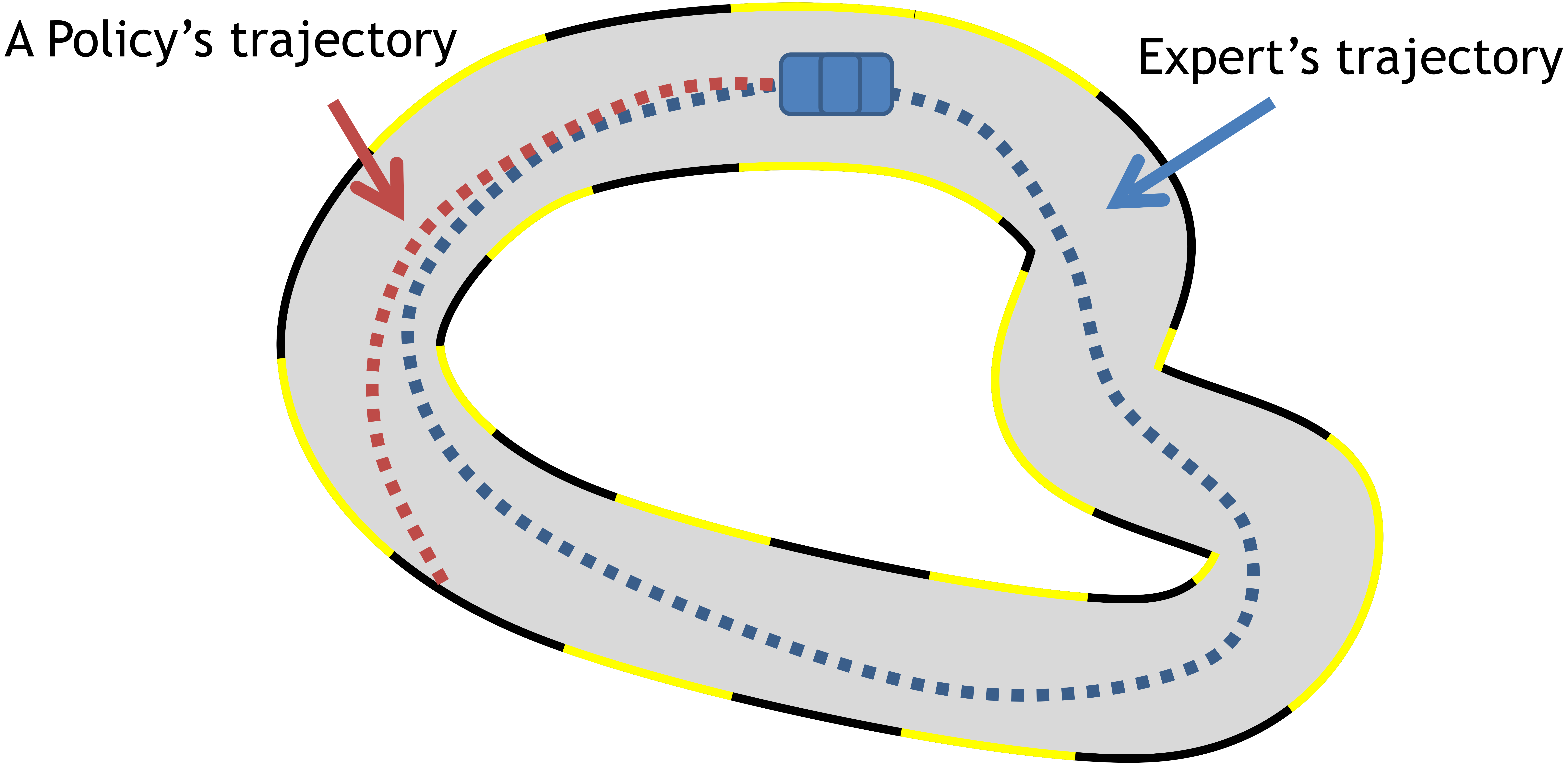
2. For performance: $V^{\pi^\star} - V^{\pi} \leq \frac{1}{1-\gamma}\left[\mathbb{E}_{s,a\sim d^{\pi^\star}} r(s,a) - \mathbb{E}_{s,a\sim d^{\widehat{\pi}}} r(s,a)\right]$

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

$$V^{\pi^\star} - V^{\widehat{\pi}} = \mathcal{O}\left(\frac{1}{(1-\gamma)^2}\sqrt{\frac{\ln(|\Pi|/\delta)}{M}}\right)$$

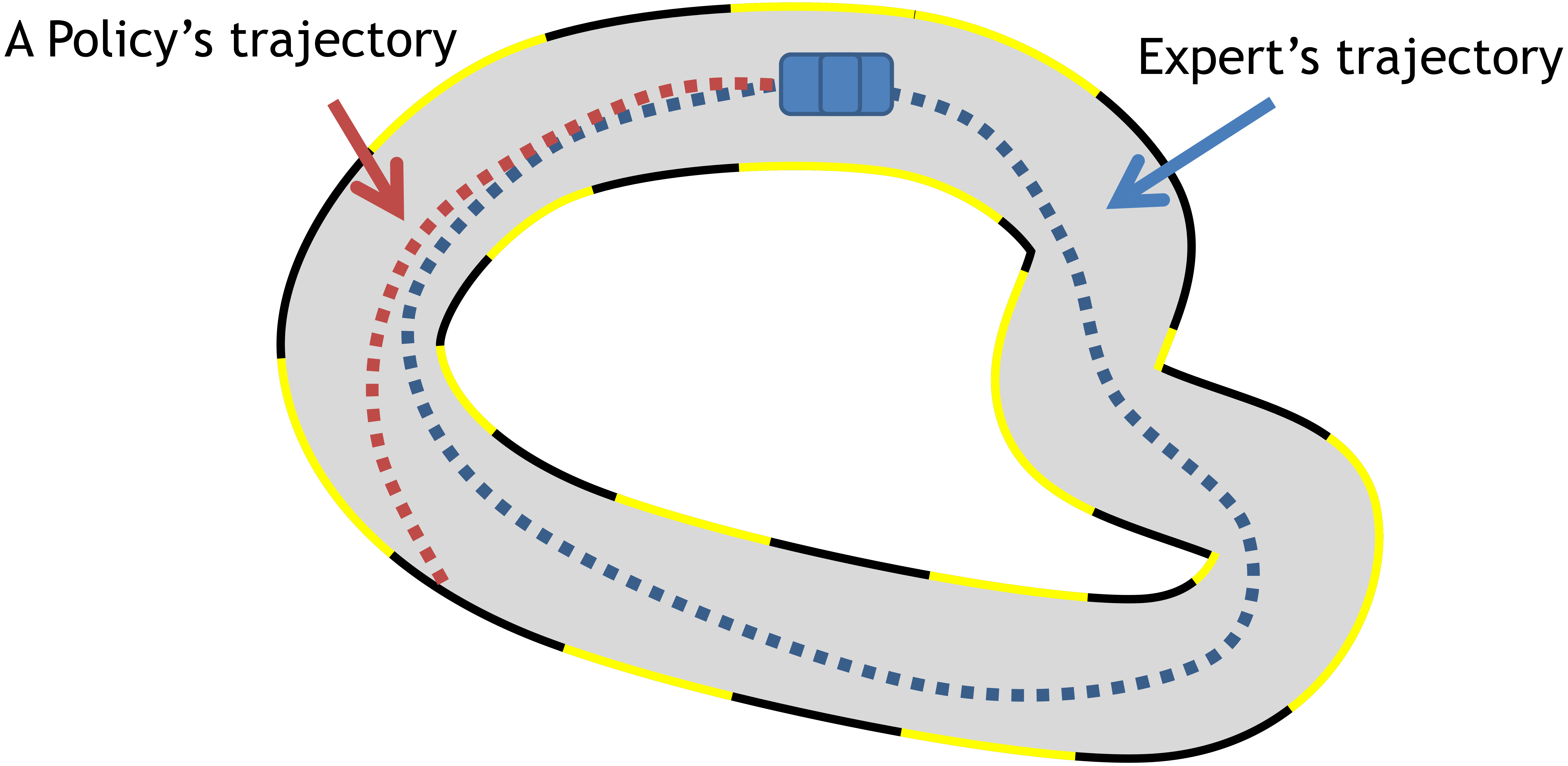

A Policy's trajectory
Expert's trajectory

d^{π} : generator that generates state-action pairs



d^{π} : generator that generates state-action pairs

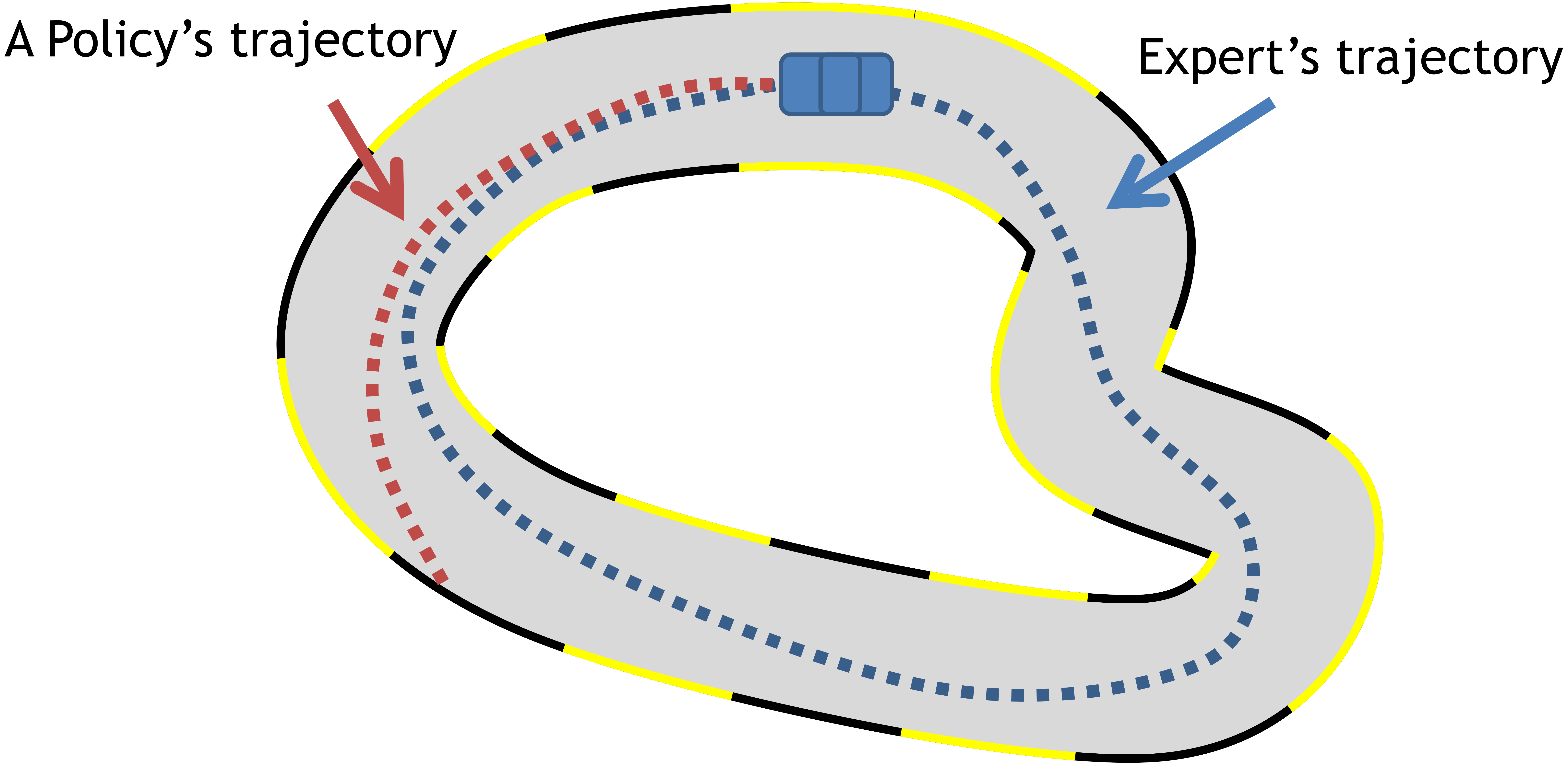
$d^{\pi^\star}$: Ground truth state-action distribution (we have samples from it)



d^{π} : generator that generates state-action pairs

$d^{\pi^\star}$: Ground truth state-action distribution (we have samples from it)

$\widetilde{\mathcal{F}}$: discriminators which distinguish red and blue



Conclusion:

1. Offline RL: only use offline expert data

BC is simple and easy to implement, has reasonable guarantees; but the quadratic dependency on horizon could cause real problems

Conclusion:

1. Offline RL: only use offline expert data

BC is simple and easy to implement, has reasonable guarantees; but the quadratic dependency on horizon could cause real problems

2. Hybrid RL: offline expert data + known transition (simulator)

Statistically, Distribution-matching has linear dependency on horizon, but the algorithm is computationally inefficient

Conclusion:

1. Offline RL: only use offline expert data

BC is simple and easy to implement, has reasonable guarantees; but the quadratic dependency on horizon could cause real problems

2. Hybrid RL: offline expert data + known transition (simulator)

Statistically, Distribution-matching has linear dependency on horizon, but the algorithm is computationally inefficient

Take home message:

There is a provable statistical benefit from the hybrid setting!

Ps: the distribution matching algorithm is very new (it was discovered when I was writing the book chapter…)