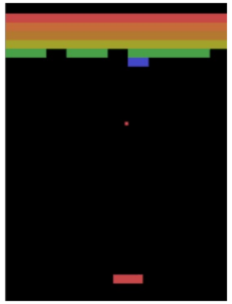
DS 598
Introduction to RL

Xuezhou Zhang

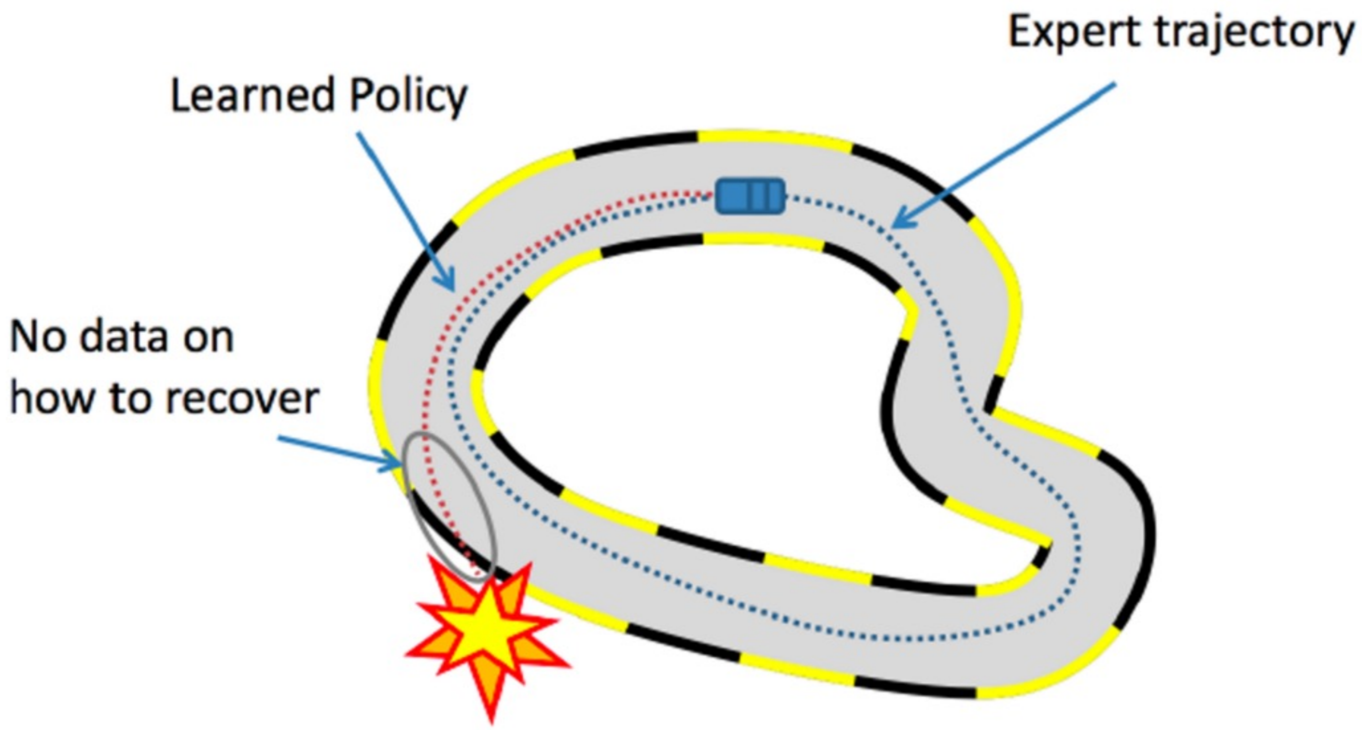
Chapter 6: Imitation Learning (Continued)

Last time: Behavior Cloning (BC)

$($  $,$  $)_{1:M}$

- Given a data set of (X, Y) pairs, predict Y as a function of X.
- This is exactly supervised learning: $\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \ell\left(\pi, s^\star, a^\star\right)$
- Only use offline expert demonstration data.

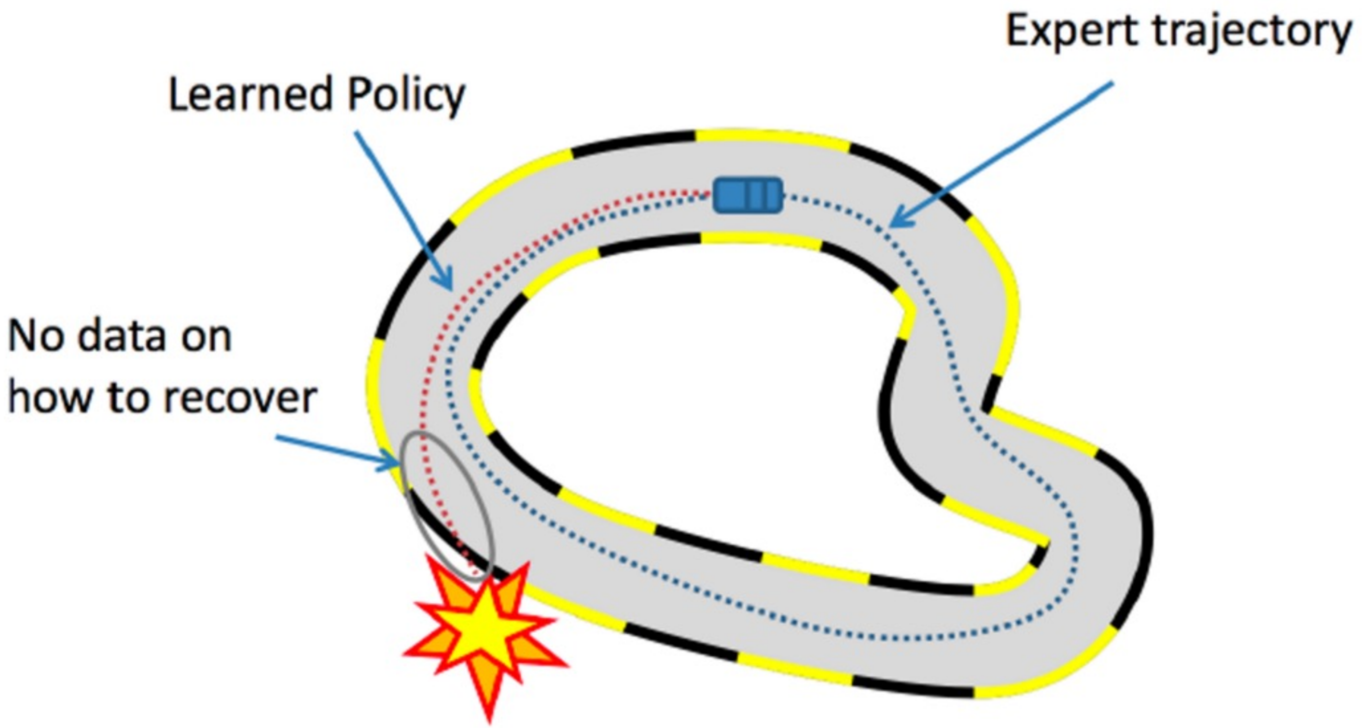
The Distribution Shift problem in BC



- This is fundamental to offline RL/IL.

How to prevent it?

- Naïve approach: expert demonstrations from all possible starting states.
- Infeasible in practice.



Today: Interactive Imitation Learning

Online Imitation Learning

- Agent interacts with the real environment.
- At any time step t, agent at (s_t, a_t).
- Agent can query $\mathrm{a}_{\mathrm{t}}^{\star} = \pi^{\star}(s_t)$ from the expert.

Dagger (Dataset Aggregation) [Ross2011]

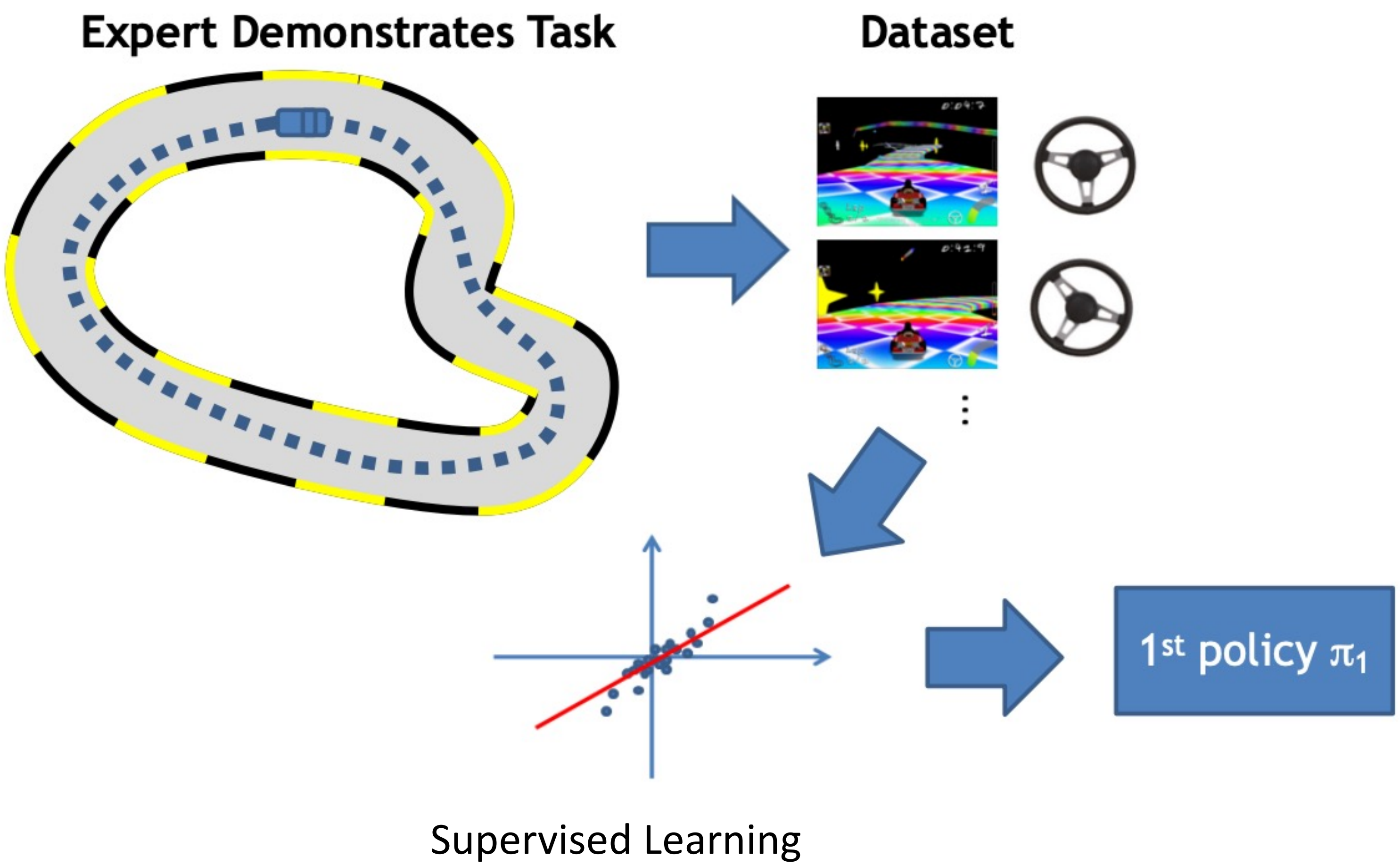
A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Stéphane Ross
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
stephaneross@cmu.edu

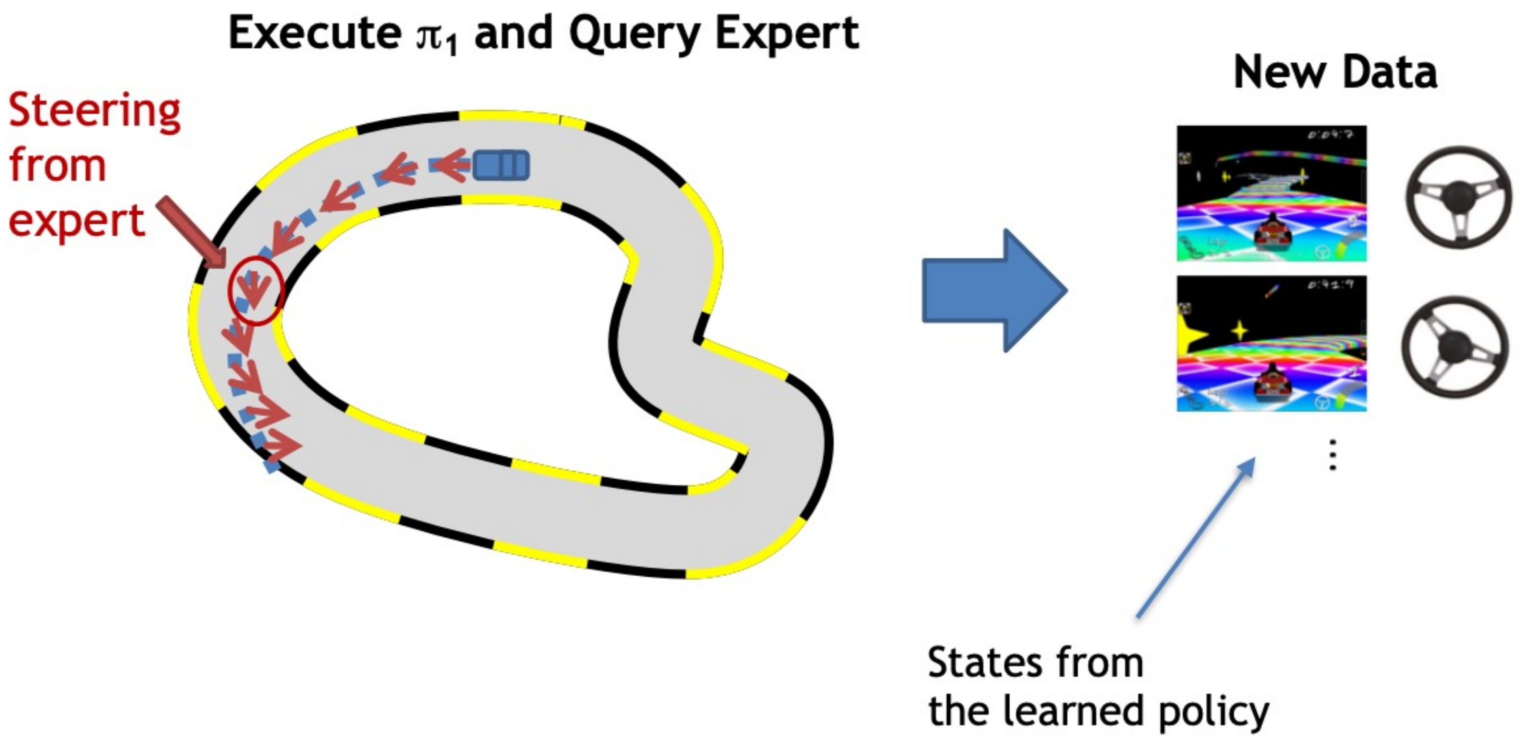
Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
ggordon@cs.cmu.edu

J. Andrew Bagnell
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dbagnell@ri.cmu.edu

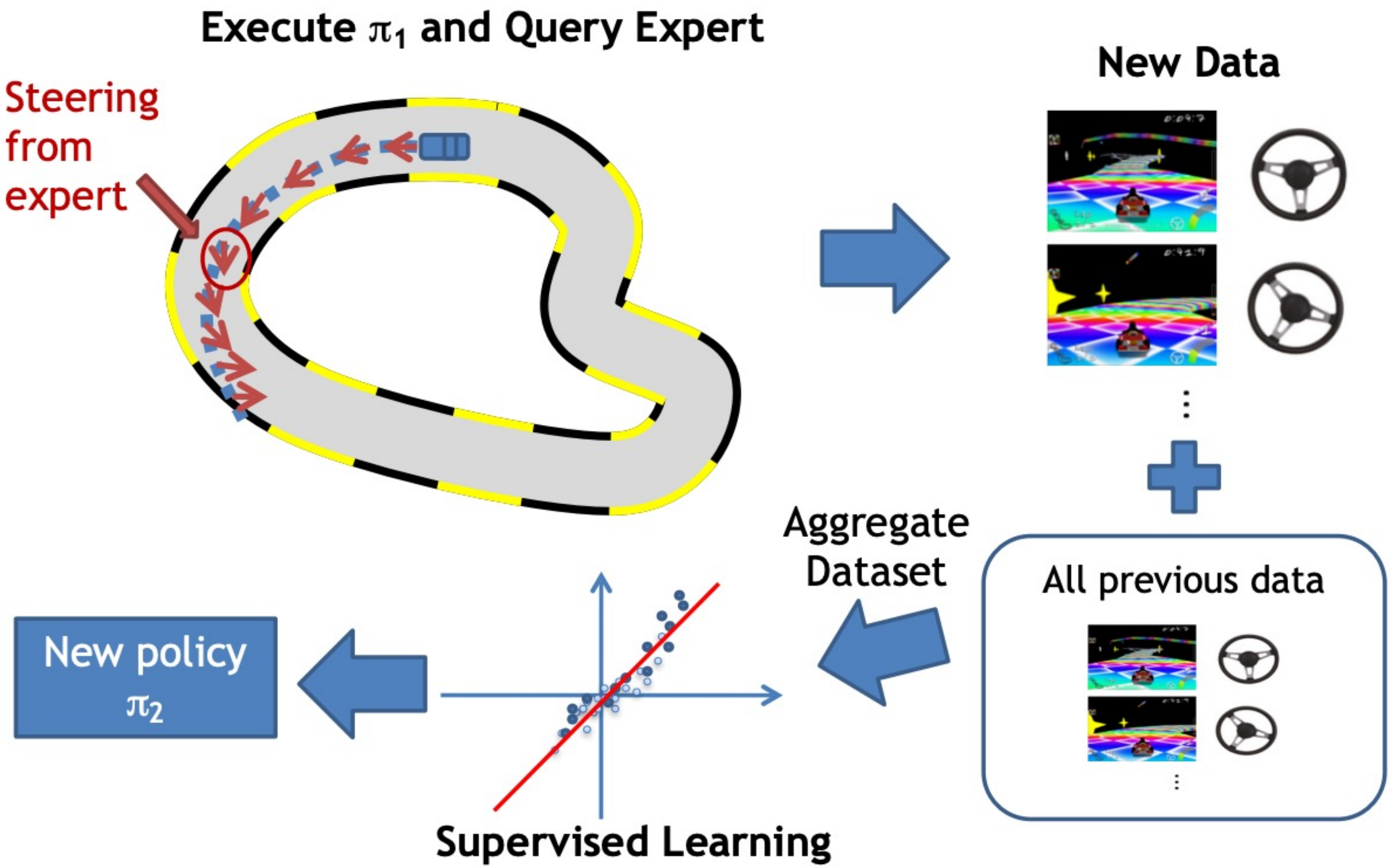
Dagger --- 0^{th} iteration



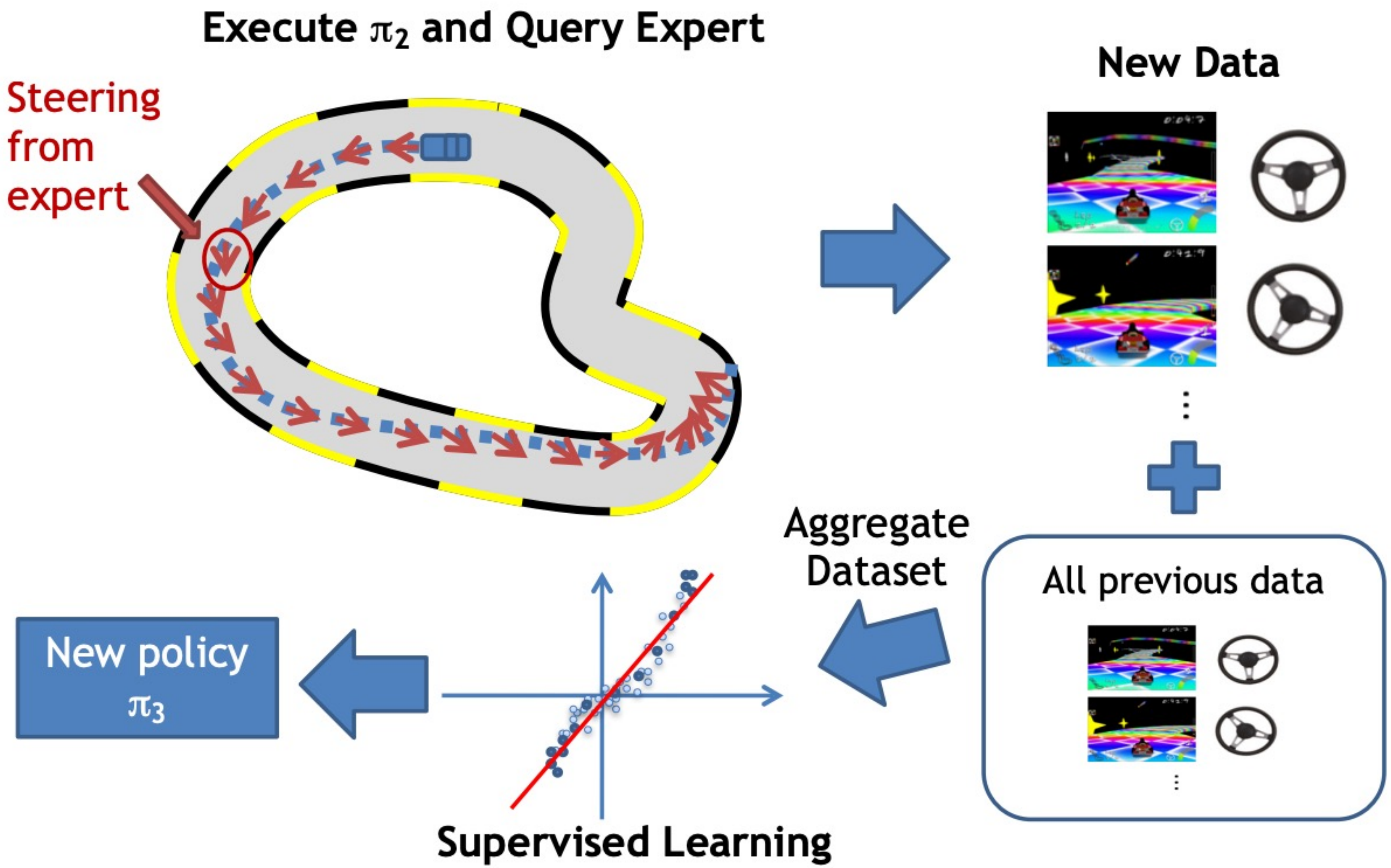
Dagger --- 1^{st} iteration



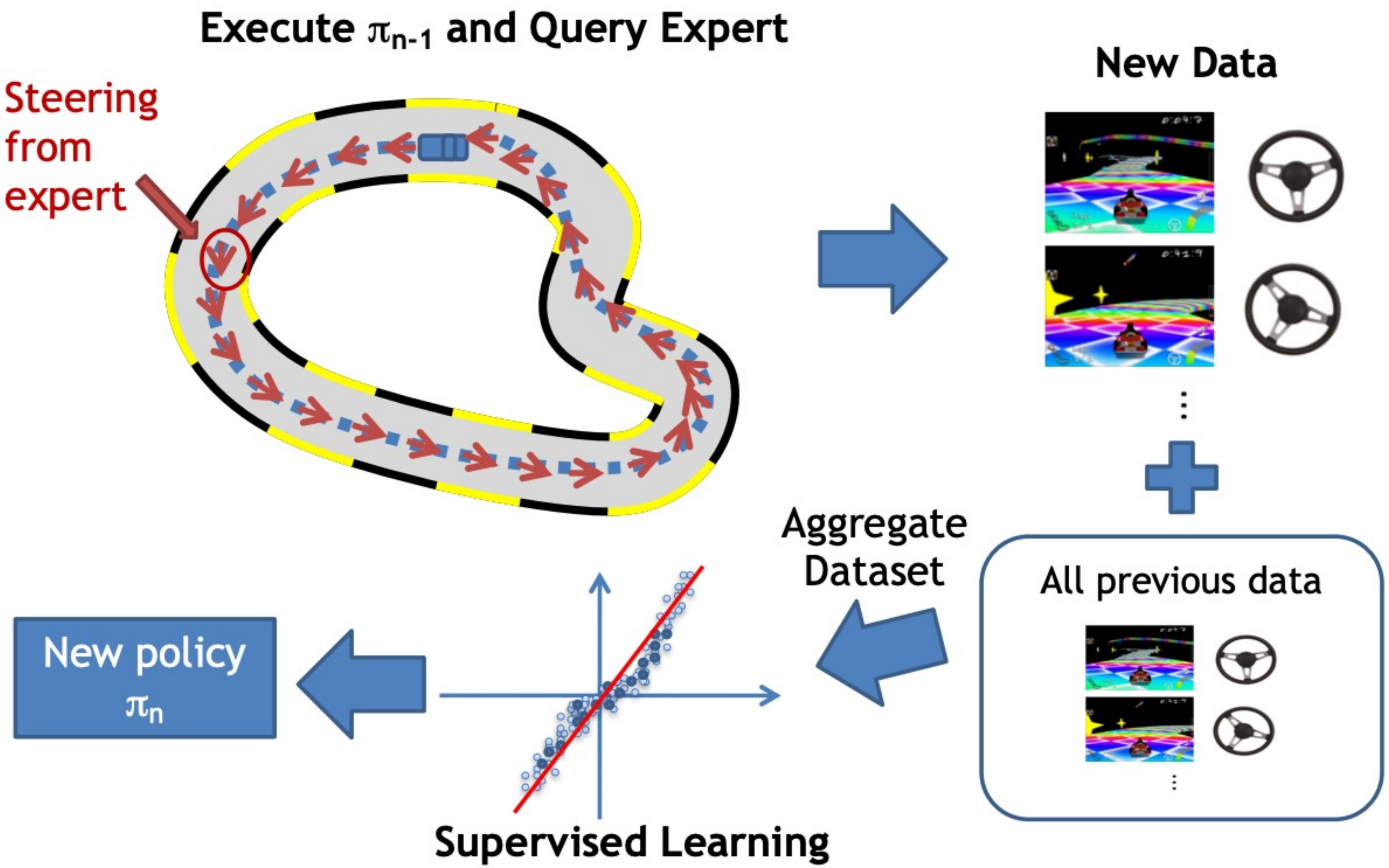
Dagger --- 2nd iteration



Dagger --- 3rd iteration



Dagger --- nth iteration



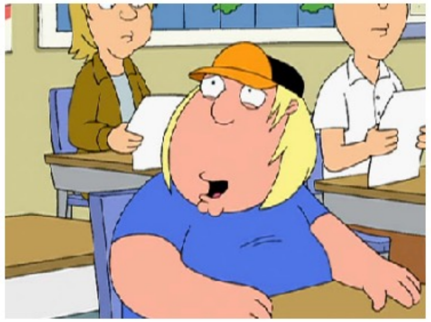
Performance of Dagger

- How do we quantify the performance of dagger?

- We need some tools from Online Learning/Online Optimization.

A Quick Intro to 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

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

Online Learning

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 $x_t \in \mathcal{X} \subset \mathbb{R}^d, y_t \in [a, b]$

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…

Online Learning

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:

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

Online Learning

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:

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

Online Learning

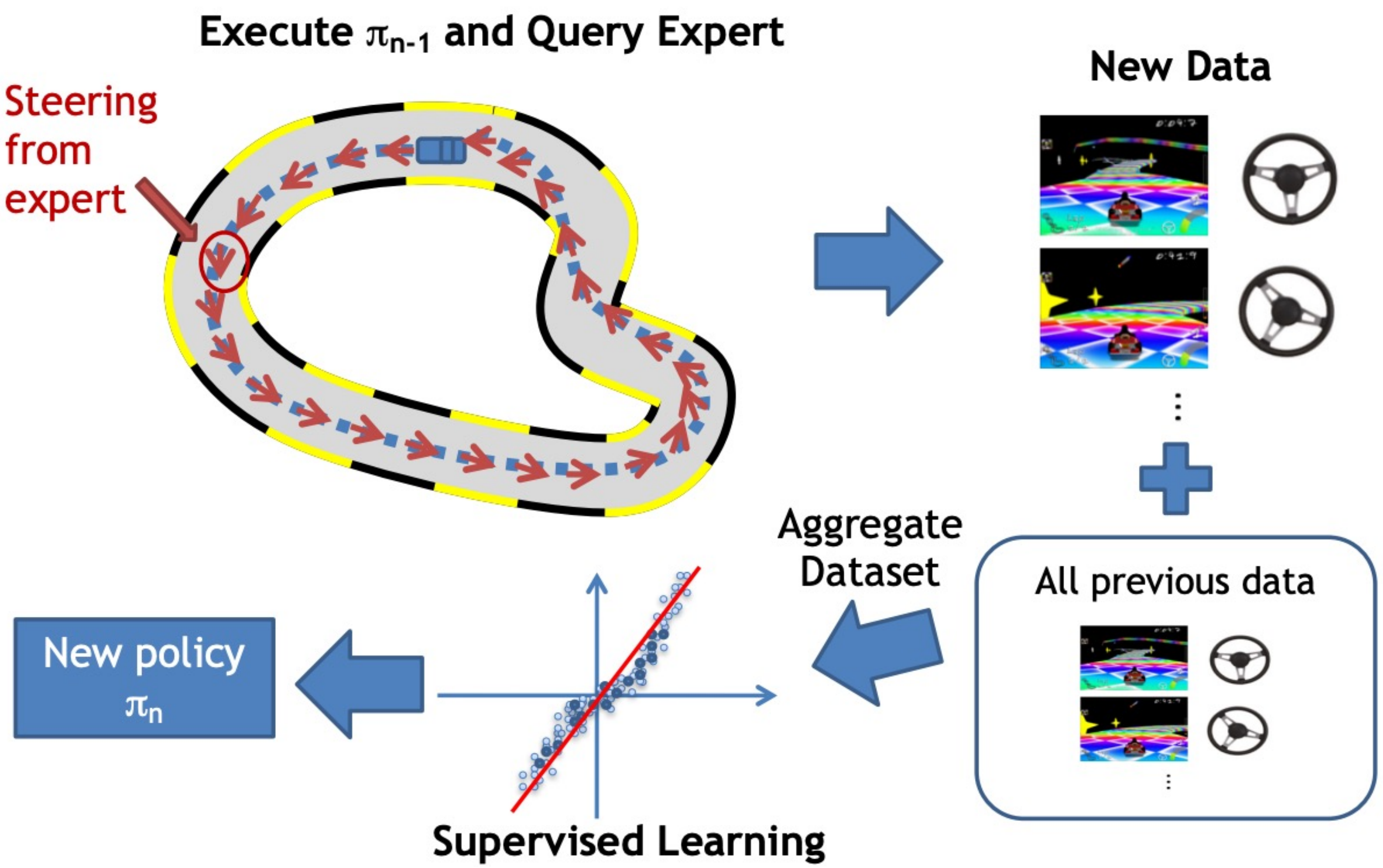
Any questions about no-regret online learning?

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

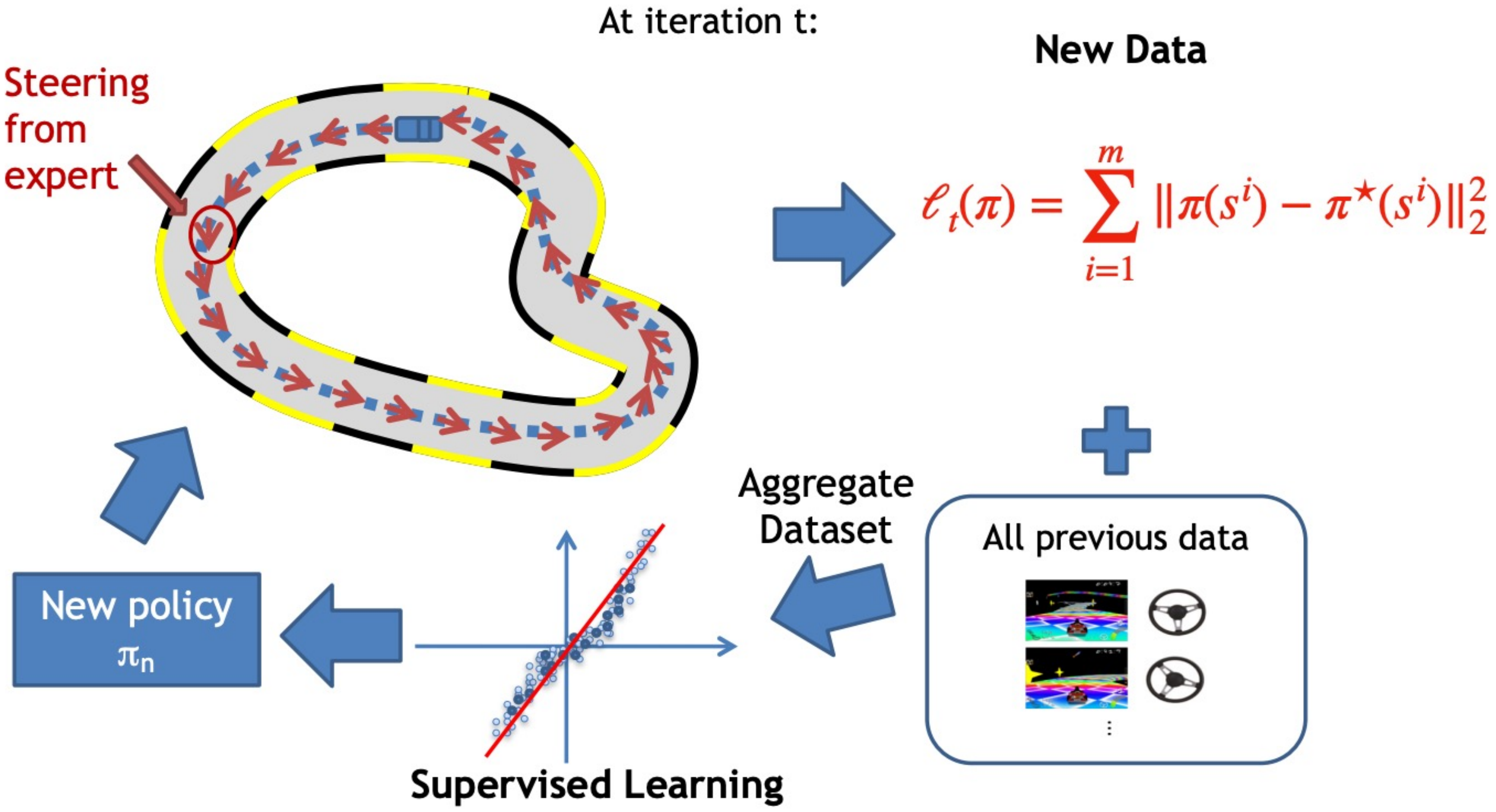
Key message:

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

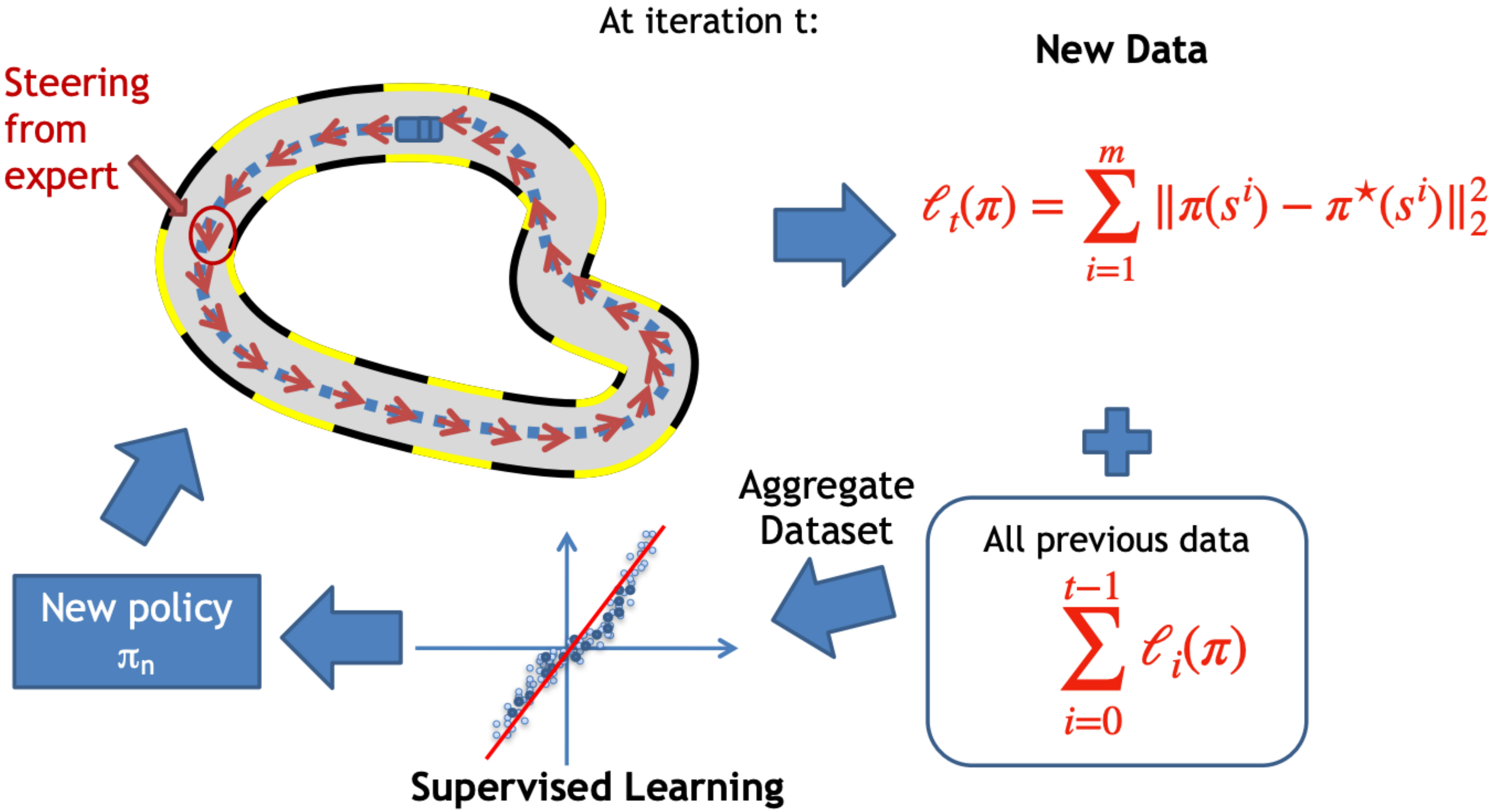
Back to Dagger



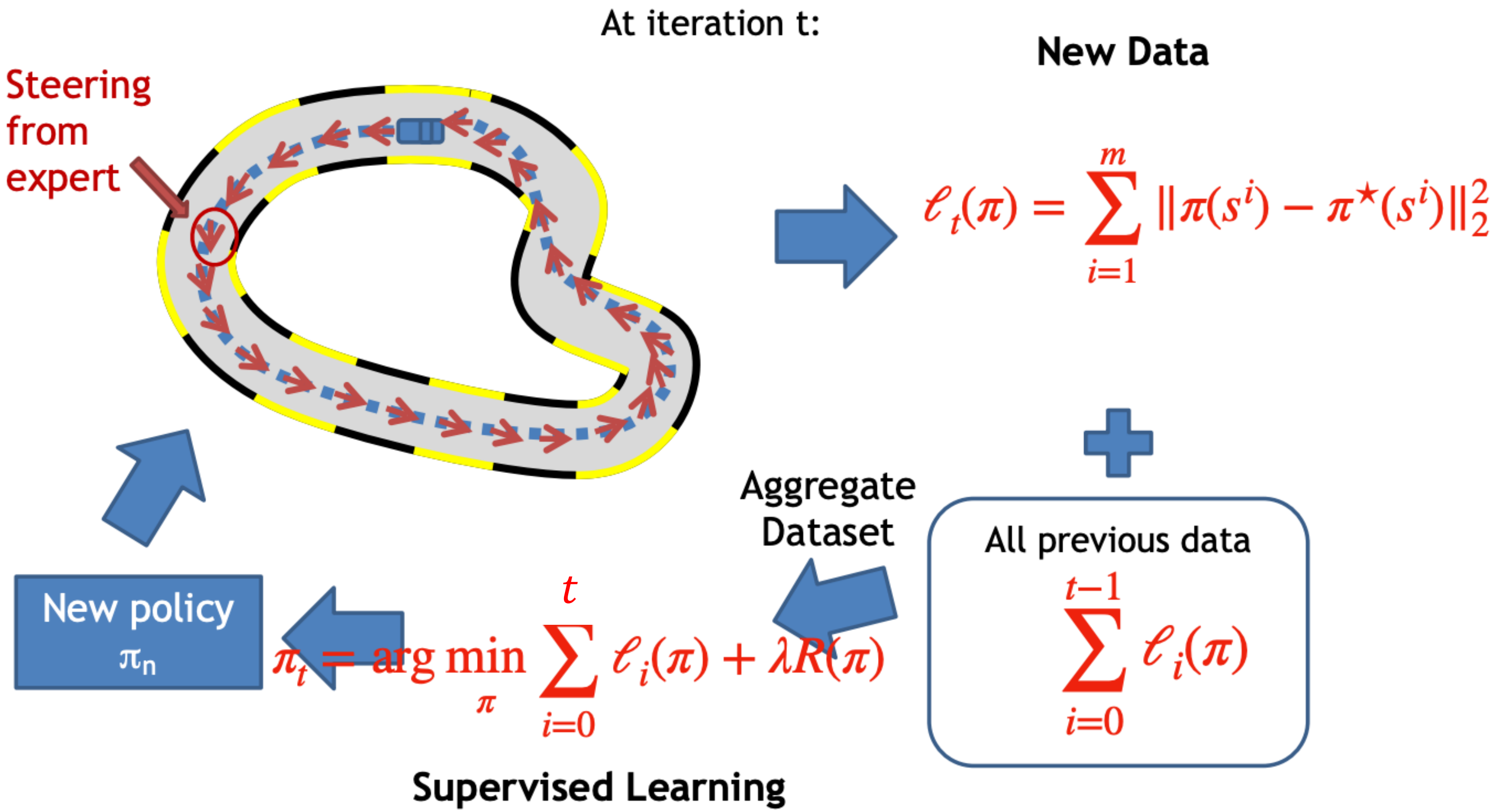
The Dagger Algorithm



The Dagger Algorithm



The Dagger Algorithm



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

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**: $\pi^{t+1} = \text{SL}\left(\mathscr{D}\right)$

- Dagger is essentially doing online learning with the SL objective.

Analysis

- Recall the online learning regret guarantee

$$\frac{1}{T}\sum_{t} \ell_t(\pi_t) - \ell_t(\pi^\star) \leq O(1/\sqrt{T})$$

- This implies, for T=$1/\epsilon^2$, there exists a $t \in [T]$, s.t.

$$\ell_t(\pi_t) - \ell_t(\pi^\star) \leq \epsilon$$

- Recall $\ell_t(\pi_t) = \mathbb{E}_{s\sim d^{\pi_t}}[1\{\pi_t(s) \neq \pi^\star(s)\}]$, so we have

$$\mathbb{E}_{s\sim d^{\pi_t}}[\mathbf{1}\{\pi_t(s) \neq \pi^\star(s)\}] \leq \epsilon$$

Recall the analysis from last time

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

$$V^{\pi^\star} - V^{\hat{\pi}} \leq \cancel{\frac{2}{(1-\gamma)^2}}\epsilon \qquad \frac{\epsilon}{1-\gamma}\max_{s,a}|A^{\pi^\star}(s,a)|$$

Proof: Performance Difference Lemma: $(1-\gamma)\big(f(\pi) - f(\pi')\big) = \mathbb{E}_{s,a\sim d^{\pi}}\big[A^{\pi'}(s,a)\big]$

$$\begin{aligned}(1-\gamma)\left(V^\star - V^{\hat{\pi}}\right) &= -\mathbb{E}_{s\sim d^{\hat{\pi}}} A^{\pi^\star}(s,\hat{\pi}(s)) \\ &\leq -\max_{s,a} A^{\pi^\star}(s,a)\mathbb{E}_{s\sim d^{\hat{\pi}}}\mathbf{1}\{\hat{\pi}(s) \neq \pi^\star(s)\} \\ &\leq \epsilon\max_{s,a}|A^{\pi^\star}(s,a)|\end{aligned}$$

"Recoverability"

We have from online learning
$\mathbb{E}_{s\sim d^{\pi_t}}[1\{\pi_t(s) \neq \pi^\star(s)\}] \leq \epsilon$

Summary

- Dagger achieves the same performance to the full coverage approach with an adaptive procedure, avoiding the quadratic blow-up.
- Problem? Online Expert query can be expensive/impossible.
- Solutions? Better HCI design. Non-human Experts.

Non-human Expert

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)

Steering + throttle

(a) raw image

Non-human Expert

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

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

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!