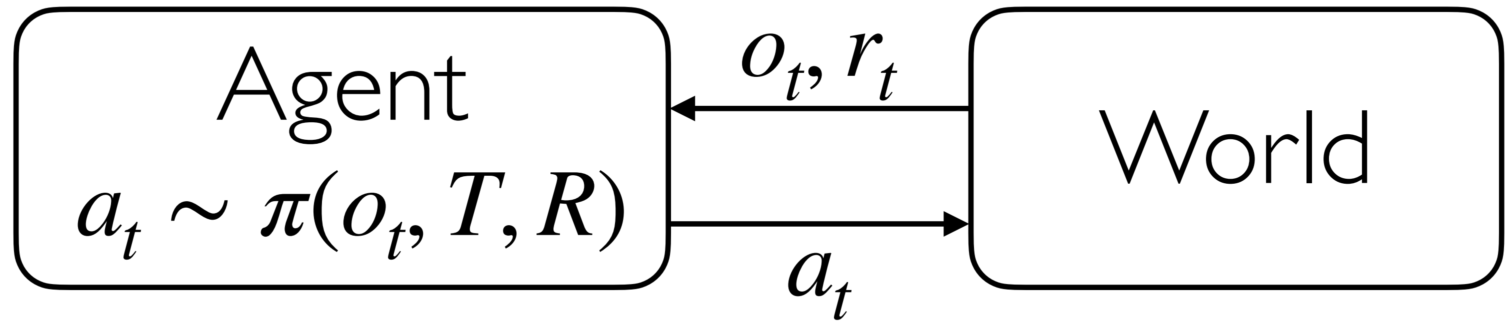
Imitation Learning

Saurabh Gupta

Convert into a Supervised Training Problem

Most General Case



Behavior Cloning

Train Time

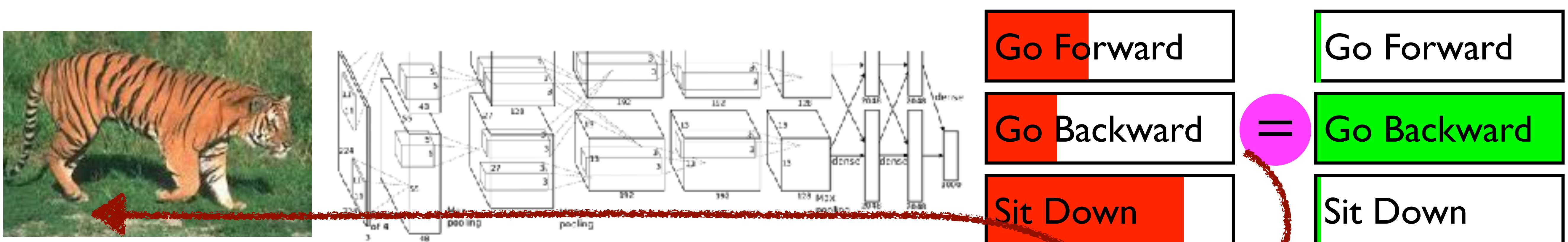
Assume an expert e can solve this MDP.

Agent $a_t \sim \pi_e(o_t)$ $\xleftarrow{o_t, r_t}$ World

Agent $\xrightarrow{a_t}$ World

1. *Ask the expert e to solve this MDP.*
2. *Collect labeled dataset D from expert.*
3. *Train a function $\pi(o_t)$ that mimics $\pi_e(o_t)$ on D.*

$\pi(o_t)$ $\pi_e(o_t)$



o_t

Train with back-propagation

Test Time

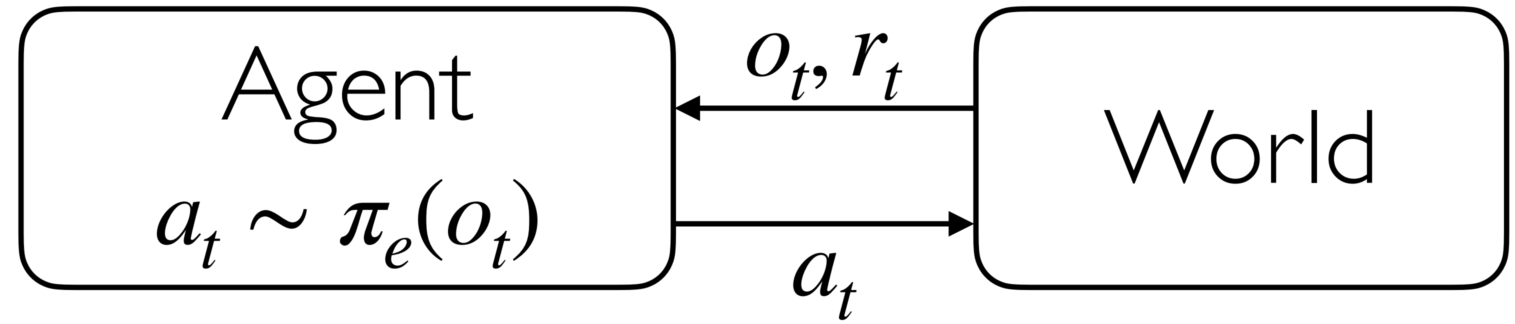
Agent $a_t \sim \pi(o_t)$ $\xleftarrow{o_t, r_t}$ World

Agent $\xrightarrow{a_t}$ World

Supervision from Human Expert

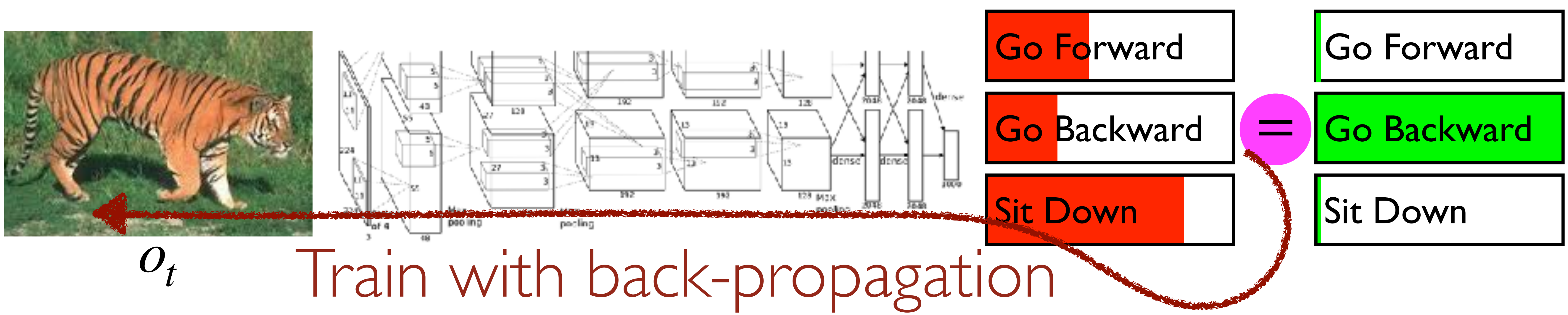
Train Time

Assume an expert e can solve this MDP.

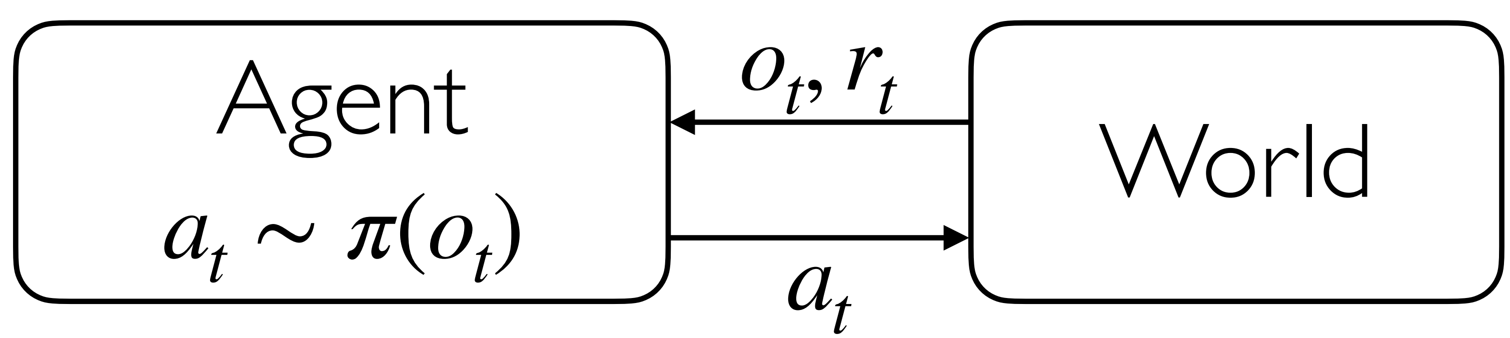


1. *Ask the expert e to solve this MDP.*
2. *Collect labeled dataset D from expert.*
3. *Train a function* $\pi(o_t)$ *that mimics* $\pi_e(o_t)$ *on D.*

$\pi(o_t)$ $\pi_e(o_t)$

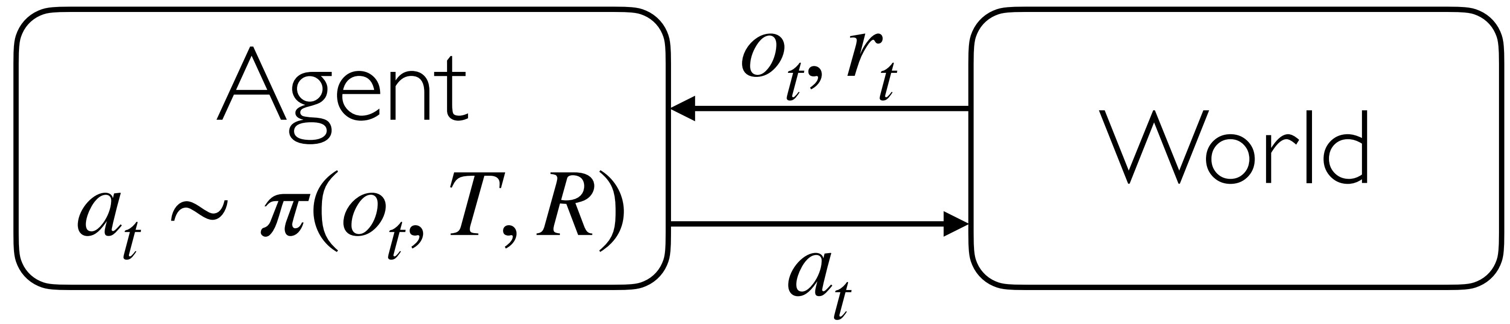


Test Time



Supervision from ~~Human~~ Algorithmic Expert

Train Time



1. Instrument the environment such that it becomes a known MDP.

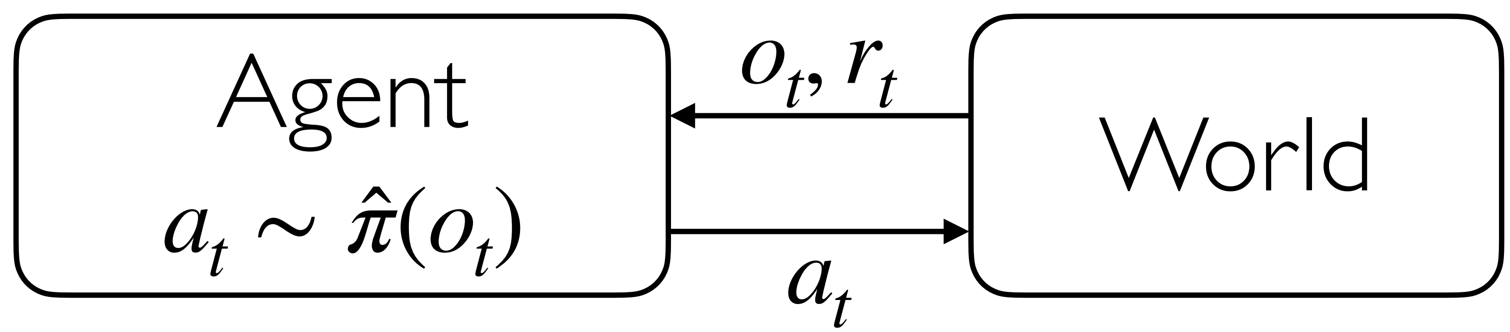
Fully Observed System $o_t = s_t$

Known or Learned Transition Function $s_{t+1} \sim \hat{T}(s_t, a_t)$

Known Reward Function $R(s_{t+1}, s_t, a_t)$

2. Train a function $\hat{\pi}(o_t)$ *that mimics* $\pi(o_t, T, R)$

Test Time

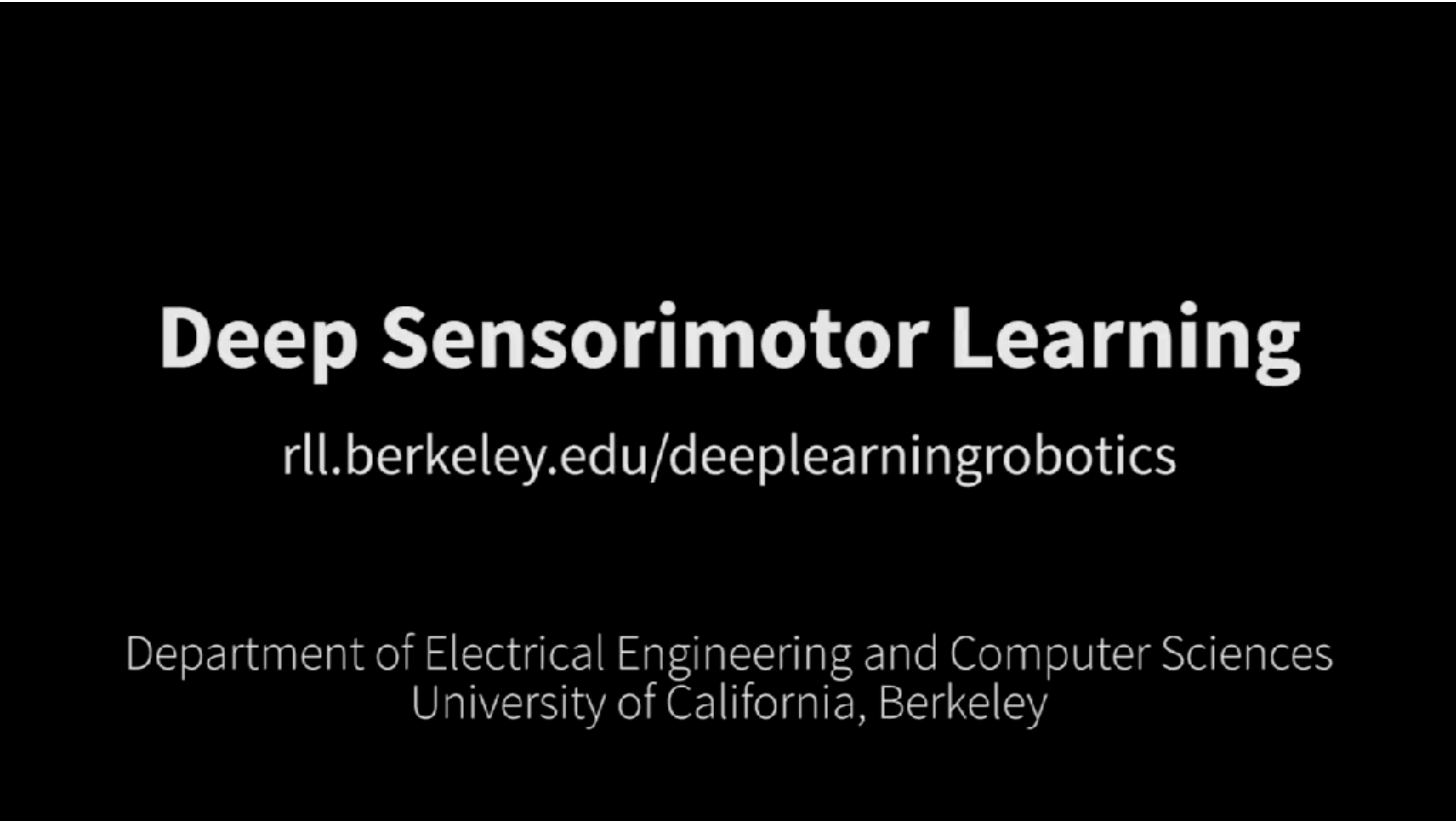


Fully Observed System $o_t = s_t$

Known or Learned Transition Function $s_{t+1} \sim \hat{T}(s_t, a_t)$

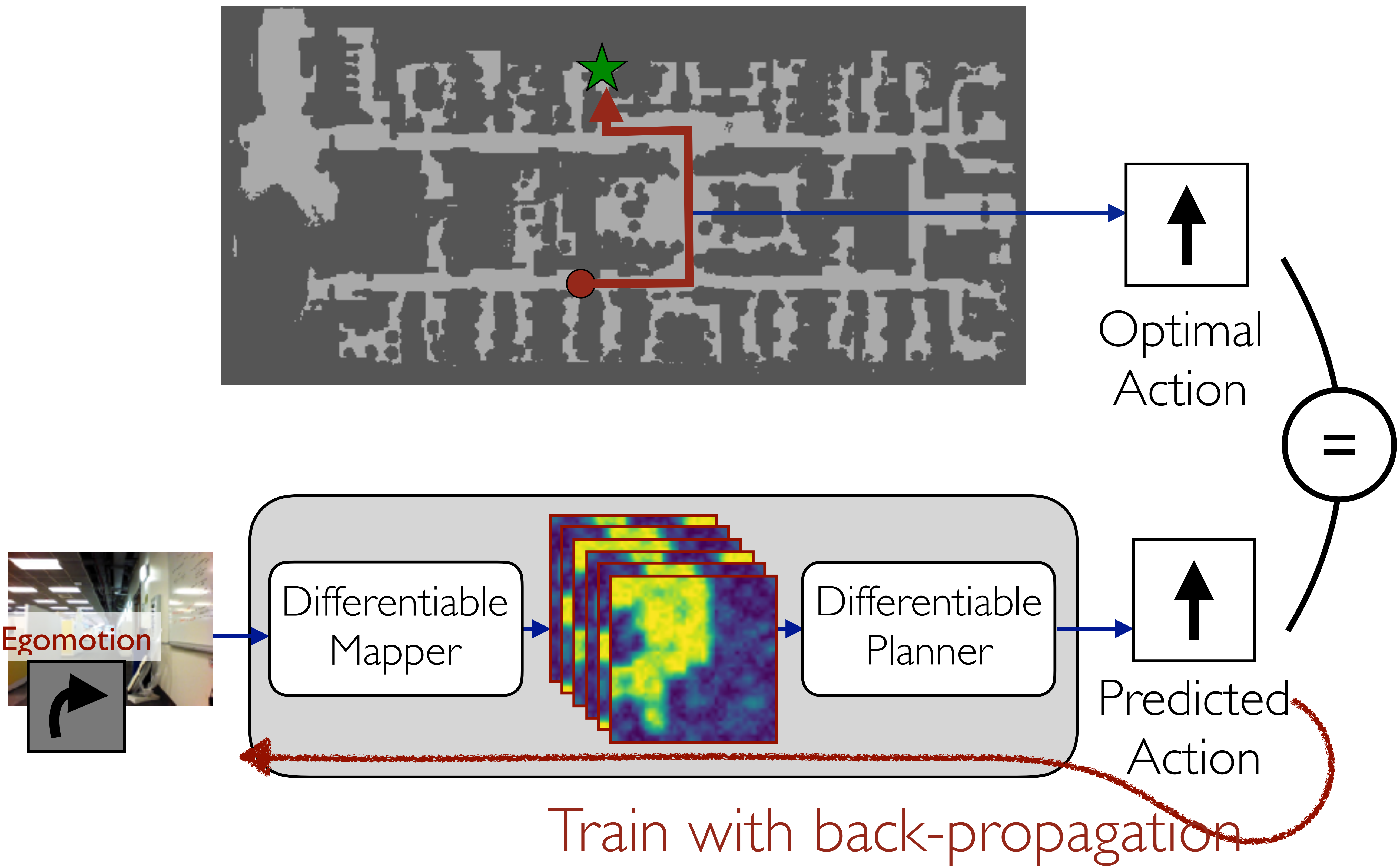
Known Reward Function $R(s_{t+1}, s_t, a_t)$

Supervision from ~~Human~~ Algorithmic Expert

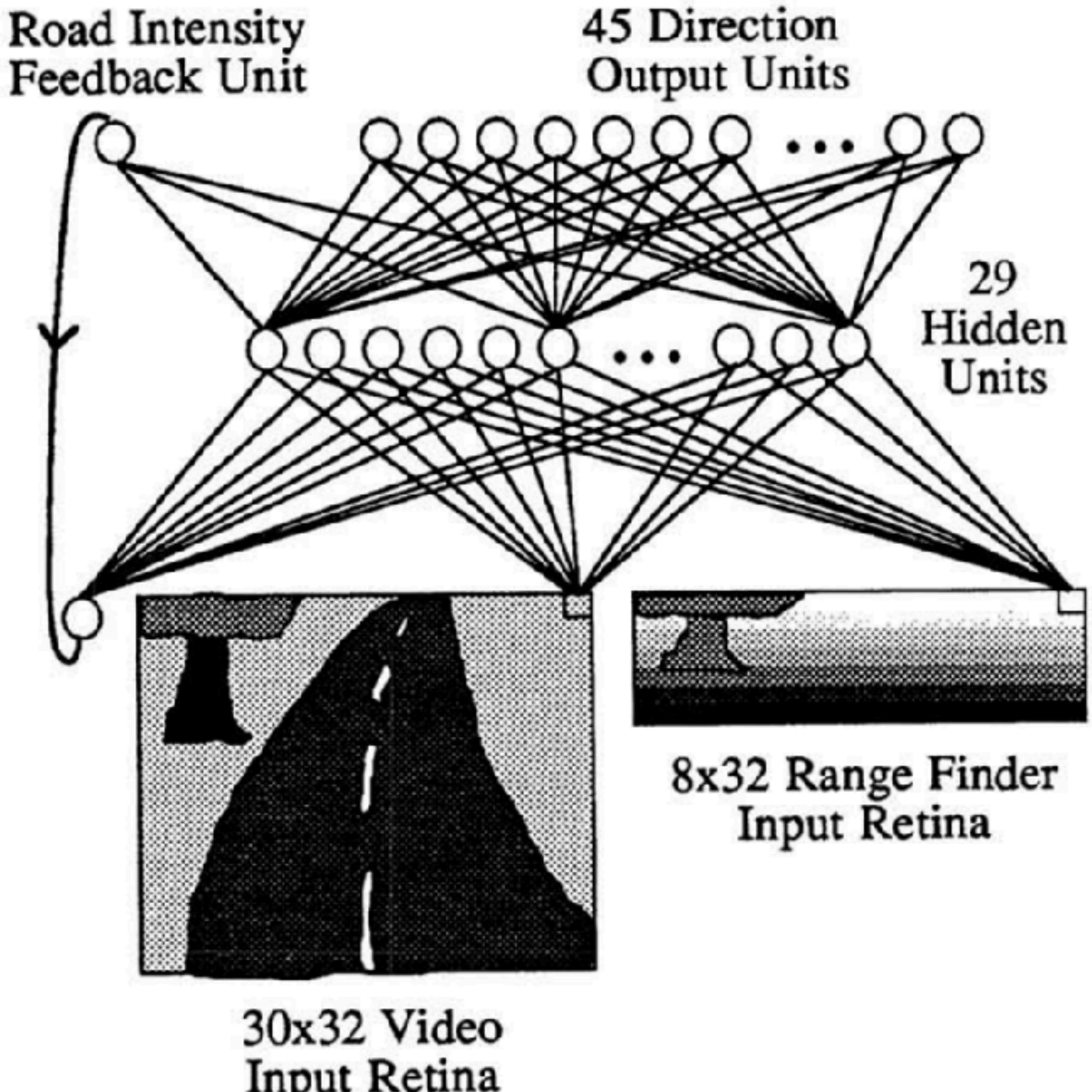


S. Levine et al. **A End-to-End Training of Deep Visuomotor Policies.** JMLR 2016.

Supervision from ~~Human~~ Algorithmic Expert



S. Gupta et al. **Cognitive Mapping and Planning for Visual Navigation.** CVPR 2017.

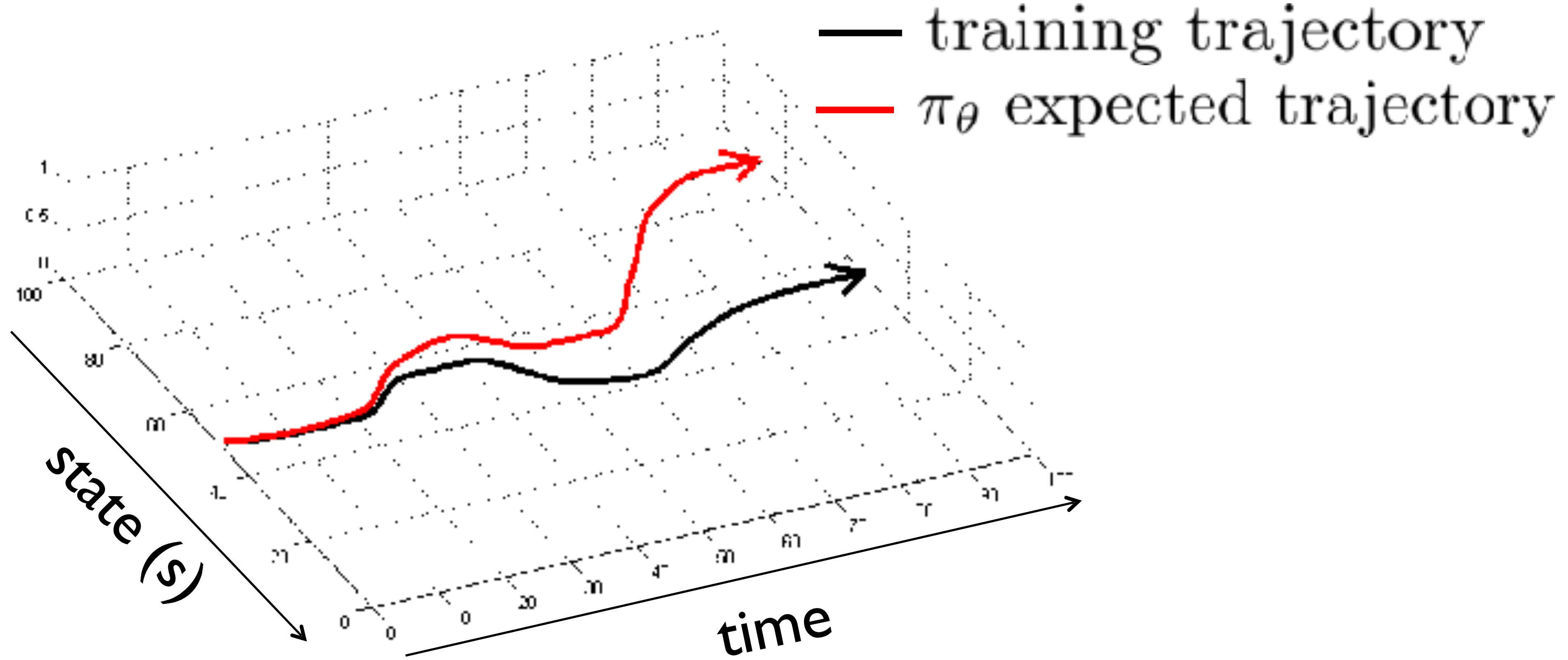


ALVINN: An Autonomous Land Vehicle in a Neural Network. Pomerleau. NeurIPS1988.

Behavior Cloning

Does it always work?

No, data mis-match problem

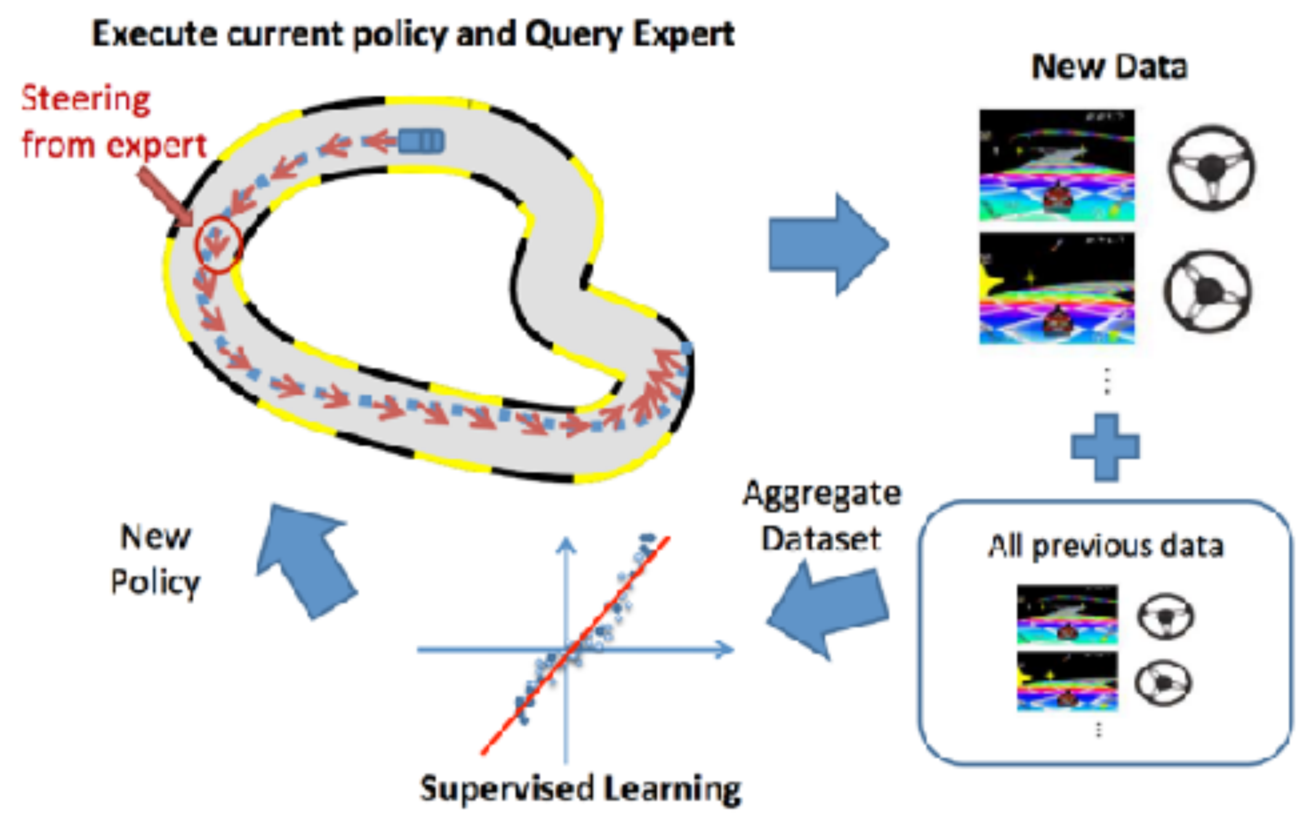


Fix Data Mis-Match Problem

DAgger: Dataset Aggregation

Collect labels on states visited by $\pi(o_t)$ *instead of* $\pi_e(o_t)$.

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



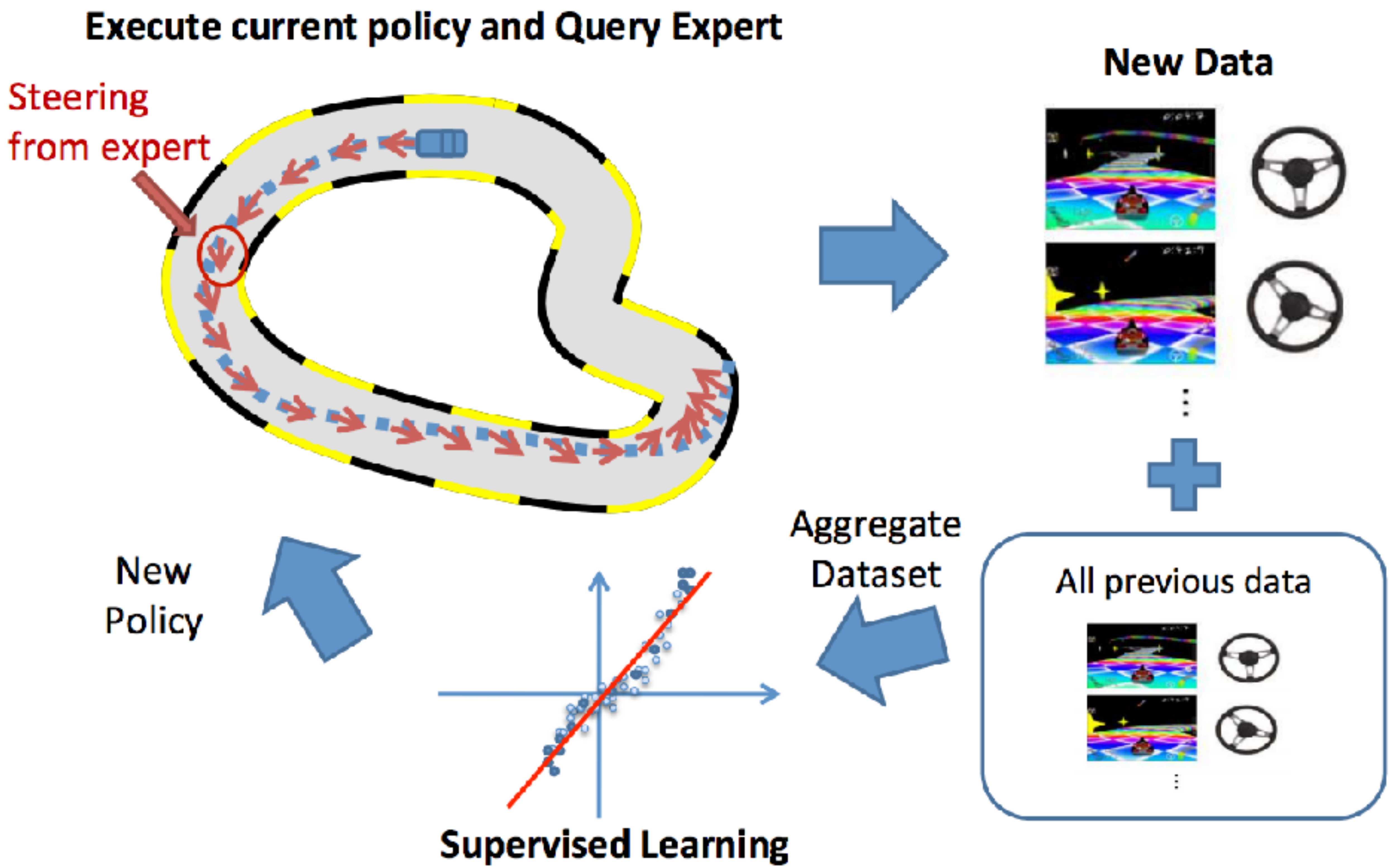
S. Ross et al. **A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning.** AISTAT 2011.

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


Execute current policy and Query Expert
Steering
from expert
New Data
Aggregate
Dataset
All previous data
New
Policy
Supervised Learning

Forward Training and SMILe Algorithm

Initialize $\pi_1^0, \ldots, \pi_T^0$ to query and execute π^*.
for $i = 1$ **to** T **do**
  Sample T-step trajectories by following π^{i-1}.
  Get dataset $\mathcal{D} = \{(s_i, \pi^*(s_i))\}$ of states, actions taken by expert at step i.
  Train classifier $\pi_i^i = \operatorname{argmin}_{\pi \in \Pi} \mathbb{E}_{s \sim \mathcal{D}}(e_\pi(s))$.
  $\pi_j^i = \pi_j^{i-1}$ for all $j \neq i$
end for
Return $\pi_1^T, \ldots, \pi_T^T$

Algorithm 3.1: Forward Training Algorithm.

Initialize $\pi^0 \leftarrow \pi^*$ to query and execute expert.
for $i = 1$ **to** N **do**
  Execute π^{i-1} to get $\mathcal{D} = \{(s, \pi^*(s))\}$.
  Train classifier $\hat{\pi}^{*i} = \operatorname{argmin}_{\pi \in \Pi} \mathbb{E}_{s \sim \mathcal{D}}(e_\pi(s))$.
  $\pi^i = (1-\alpha)^i \pi^* + \alpha \sum_{j=1}^{i} (1-\alpha)^{j-1} \hat{\pi}^{*j}$.
end for
Remove expert queries: $\tilde{\pi}^N = \frac{\pi^N - (1-\alpha)^N \pi^*}{1-(1-\alpha)^N}$
Return $\tilde{\pi}^N$

Algorithm 4.1: The SMILe Algorithm.

DAgger

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Algorithm 3.1: DAGGER Algorithm.

Super Tux Kart

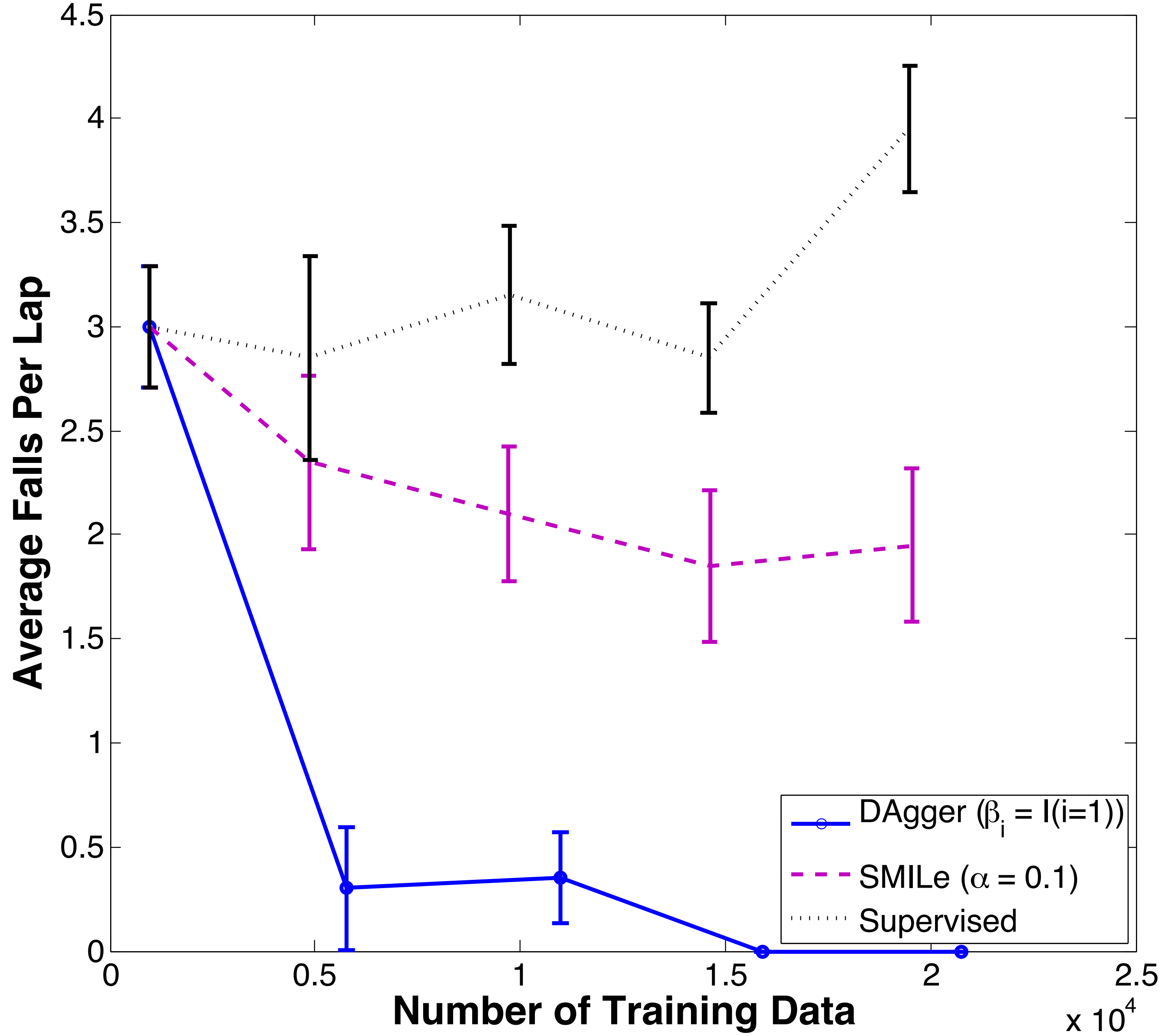


Figure 2: Average falls/lap as a function of training data.

Super Mario Bros.

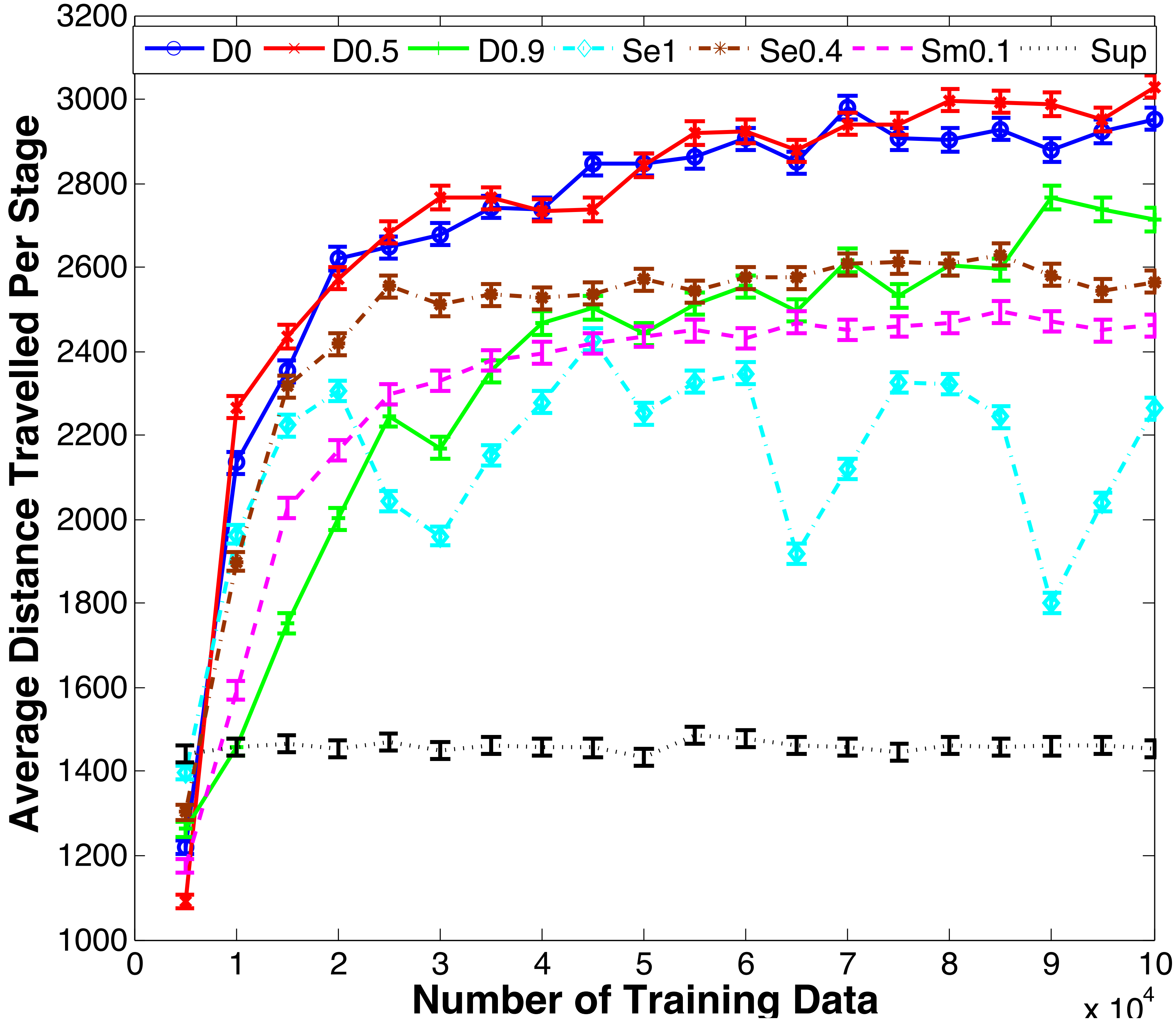


Figure 4: Average distance/stage as a function of data.

Structured Prediction

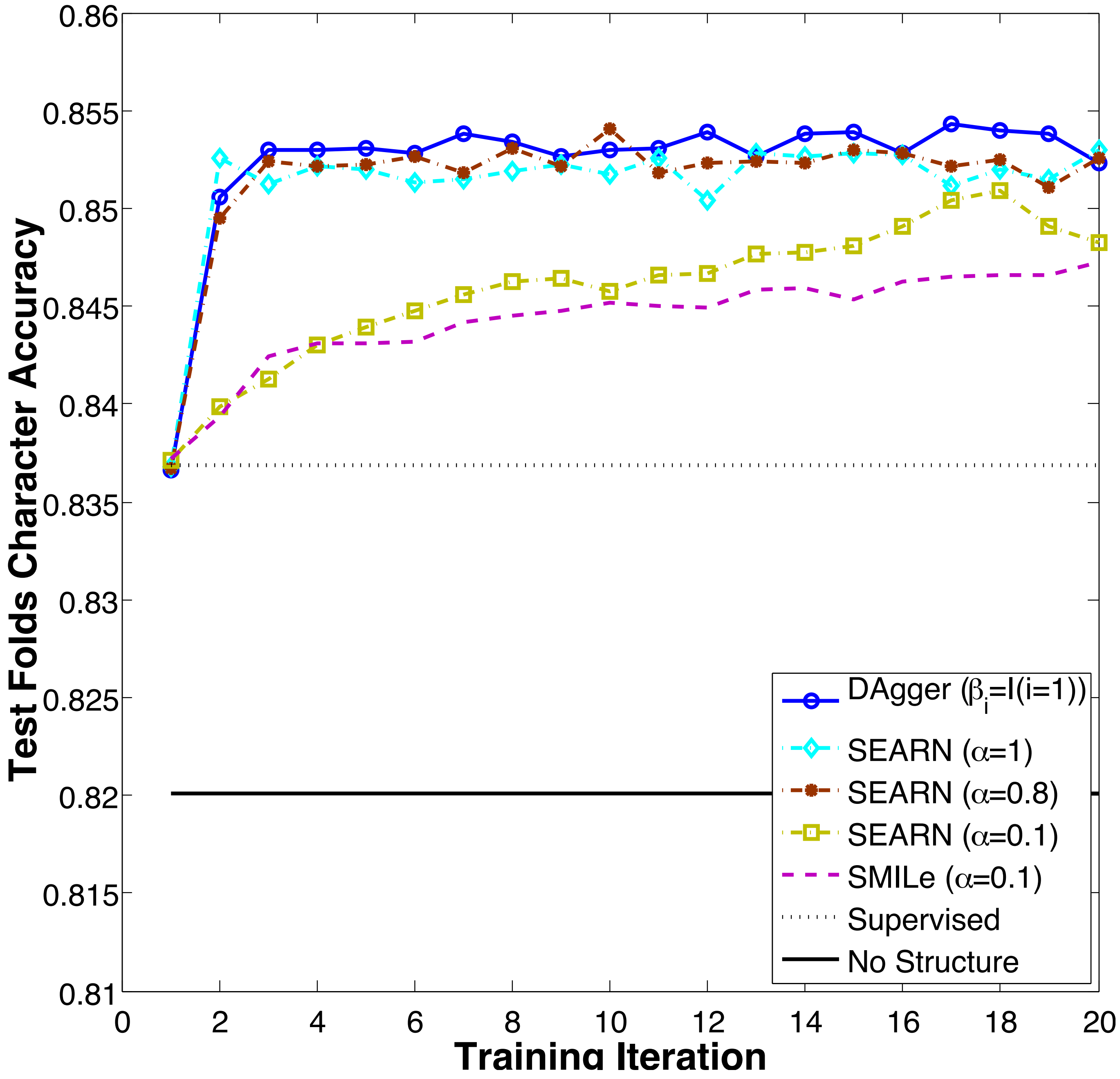


Figure 5: Character accuracy as a function of iteration.

Thank you