1. Policy Evaluation
(Online Bellman Residual)

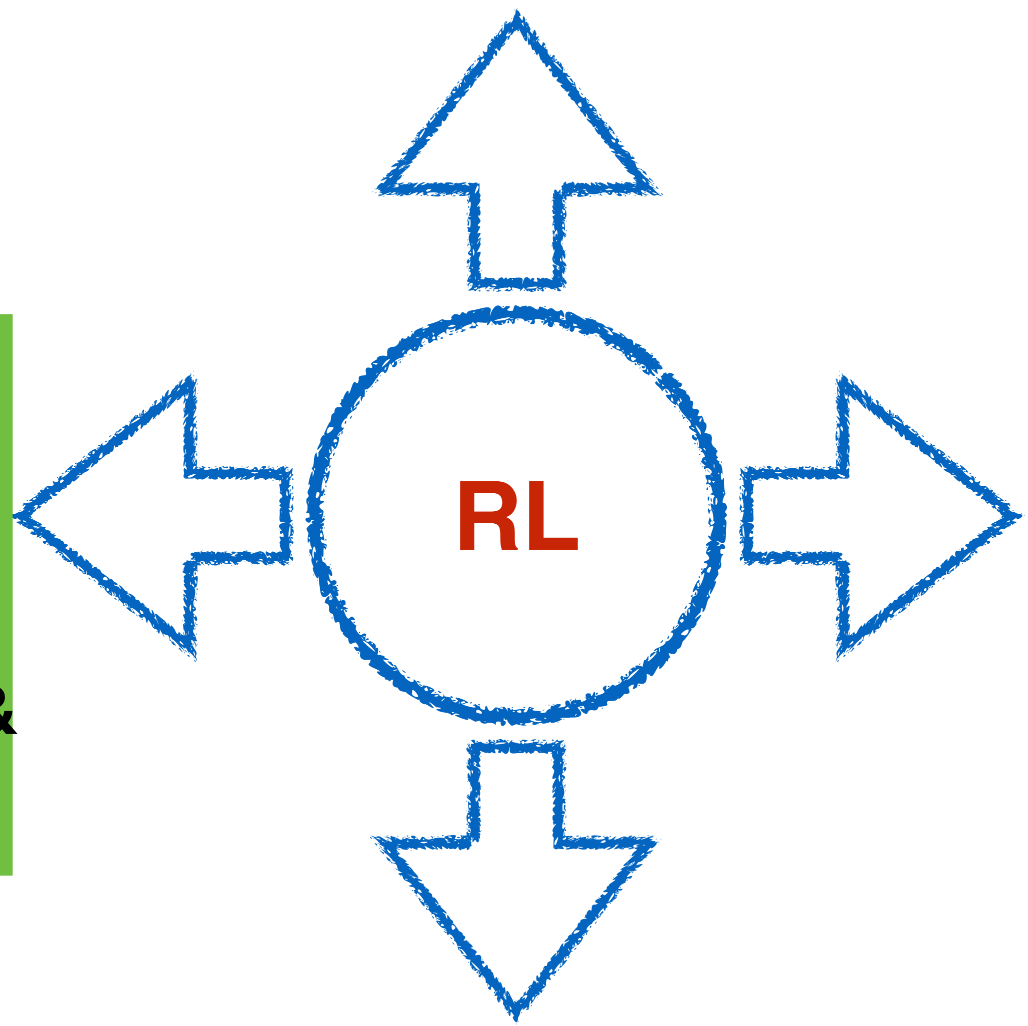
[Sun & Bagnell, 15, UAI (Best Student Paper)]

Function Approximation

2. RL via Imitation
(Imitation Learning)

[Sun et.al 17, ICML; 18, ICLR]

Function Approximation &
Imitation



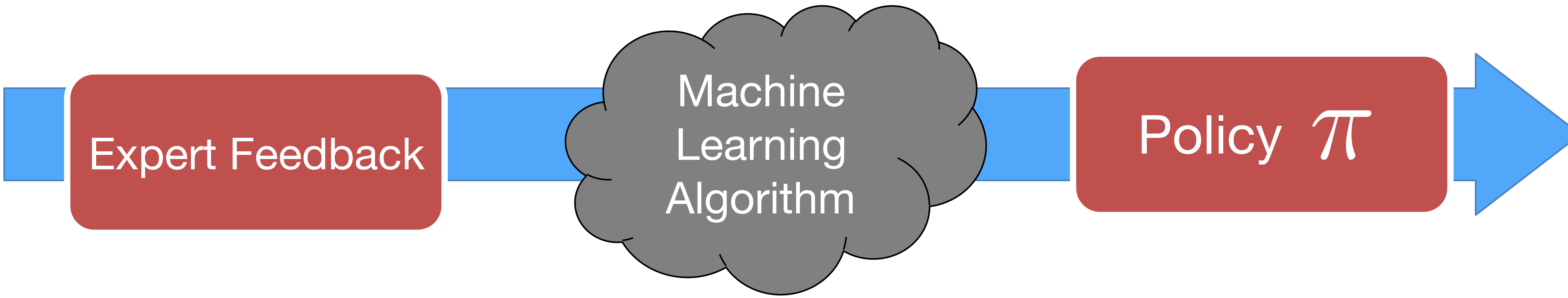
3. RL via Indirect
Imitation
(Dual Policy Iteration)

[Sun et.al, 18, submitted to ICML]

Function Approximation
Optimal Control

4. Proposed Work:
Temporal Difference Learning &
Apprenticeship Learning

Imitation Learning

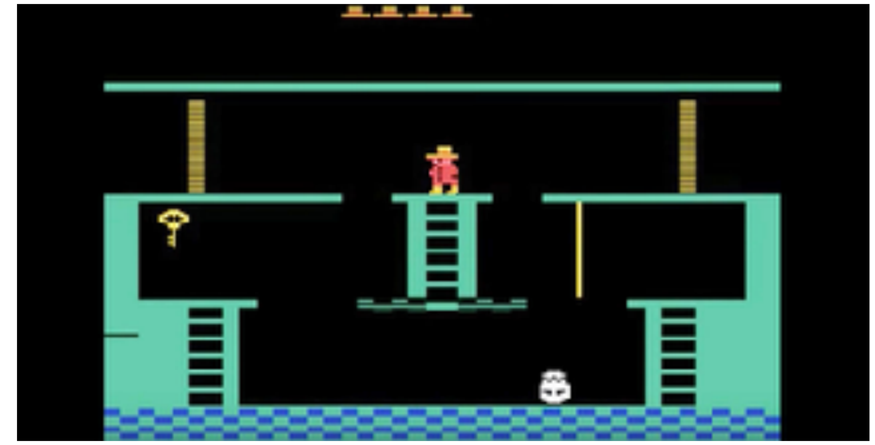


- SVM
- Gaussian Process
- Kernel Estimator
- Deep Networks
- Random Forests
- LWR
- …

Maps *states* to actions

Efficient RL via Imitation

Human



Extra Assumptions:

expert policy π^e **a**  **ining**;

cost (reward) signal

Planner & Controller
(robotics)

(e.g., Optimal Planner, MPC)
[Choudhury, et.al, ICRA, RSS, 17, Pan et.al,17]

Ground Truth
Labels + Utility
(NLP)

Search Algorithm
(e.g. A*) as expert

Why bother imitating when you have cost/reward signals?

Formalizing Advantages

ADVANTAGE #1

Less sensitive to local optimality

DAgger (Data Aggregation) [Ross et.al, 11, AISTATS]

AggreVaTe (Aggregate with Values) [Ross&Bagnell14, arxiv]

ADVANTAGE #2

More Sample Efficient (i.e., Learns faster)

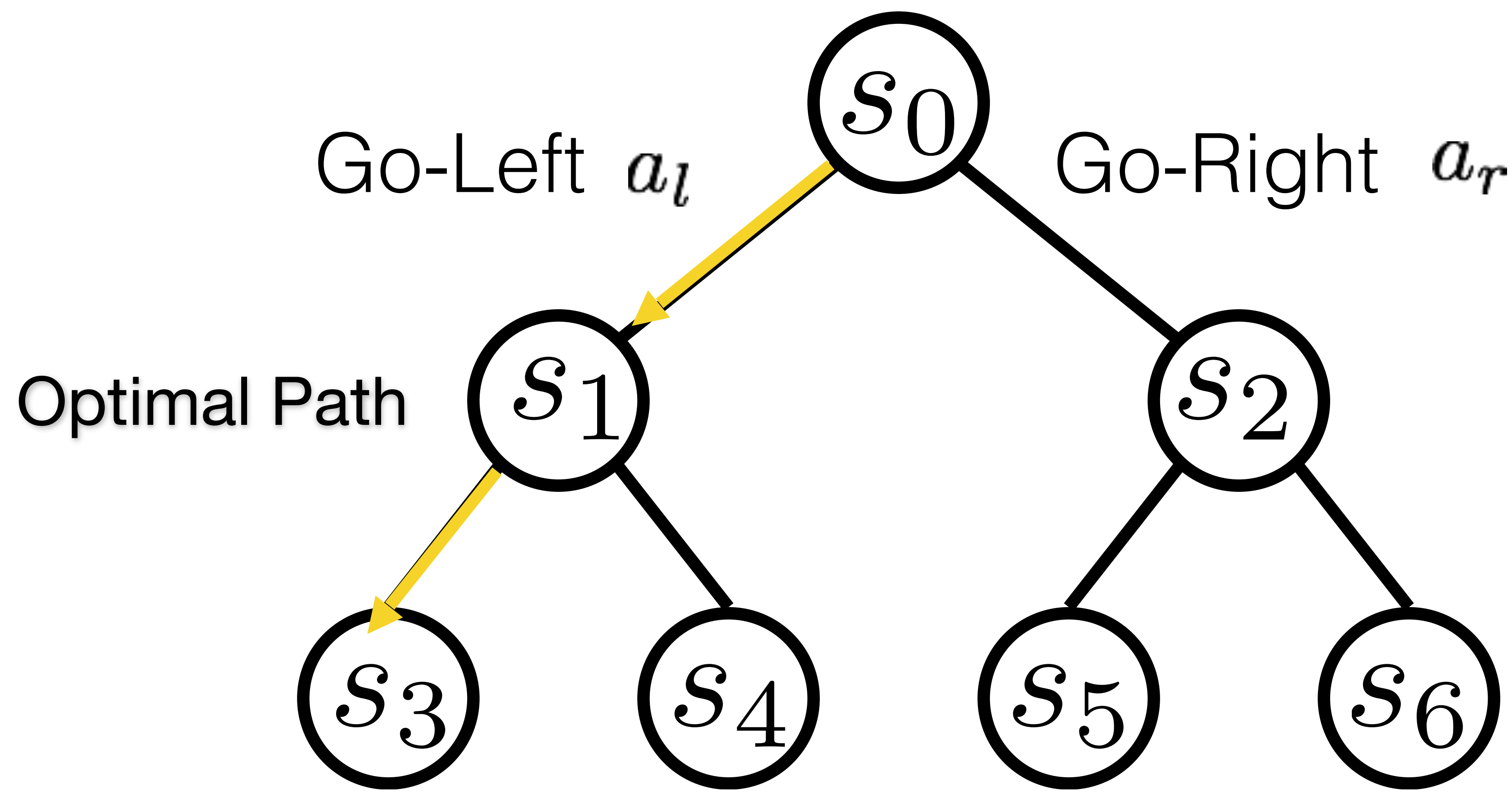
There exist problems, s.t. with access to optimal expert, i.e., $\pi^e = \pi^*$

IL learns **exponentially** faster than RL

[Sun et.al, 17,ICML]

Consider a simple, tree like MDP

let us assume $\pi^e = \pi^*$, and we can query $Q^*(s,a)$



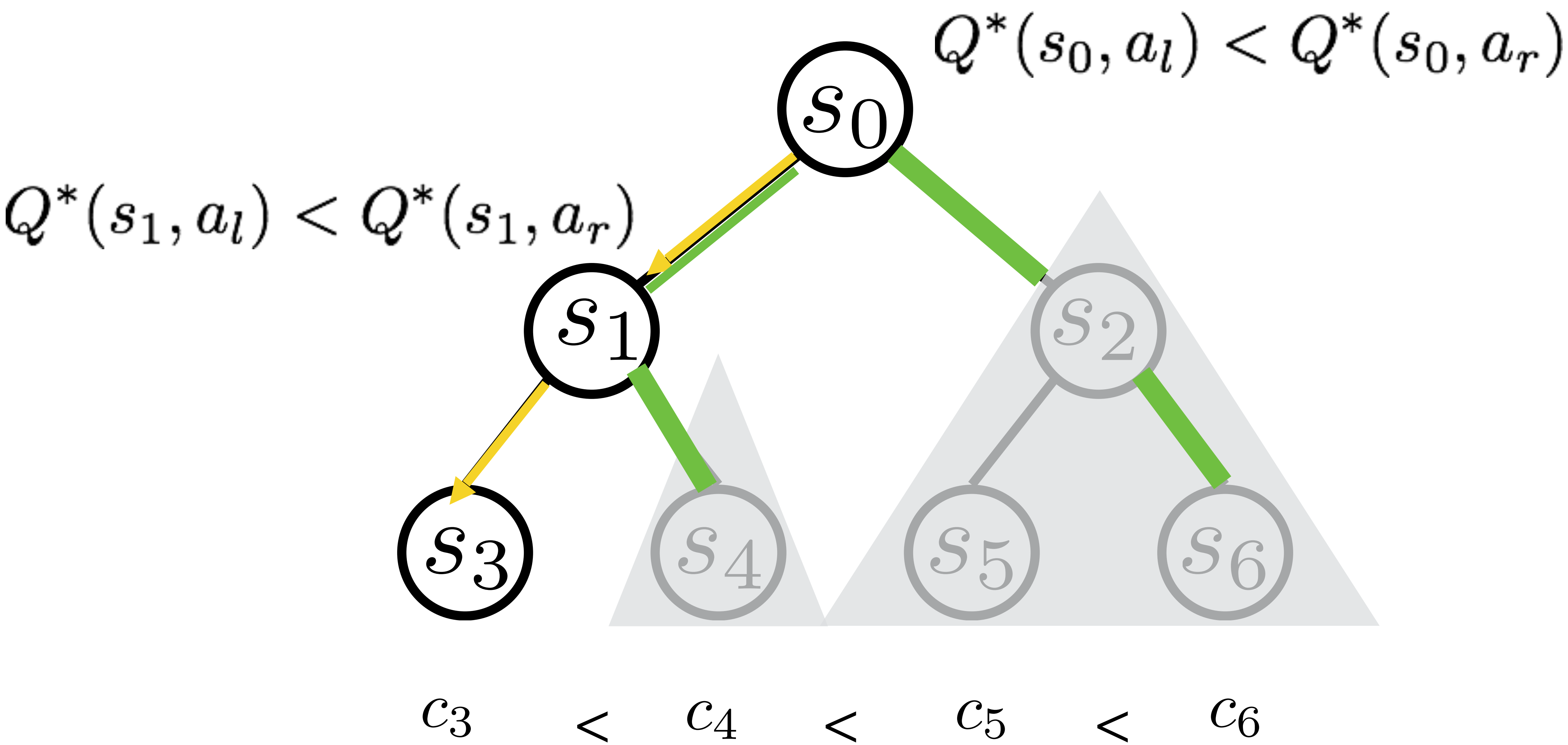
$c_3 \quad < \quad c_4 \quad < \quad c_5 \quad < \quad c_6$

$$c_{non-leaf} = 0$$

$Q^\pi(s,a)$: the total cost of taking action a at s then following policy π

And an expert who tells us which action is better

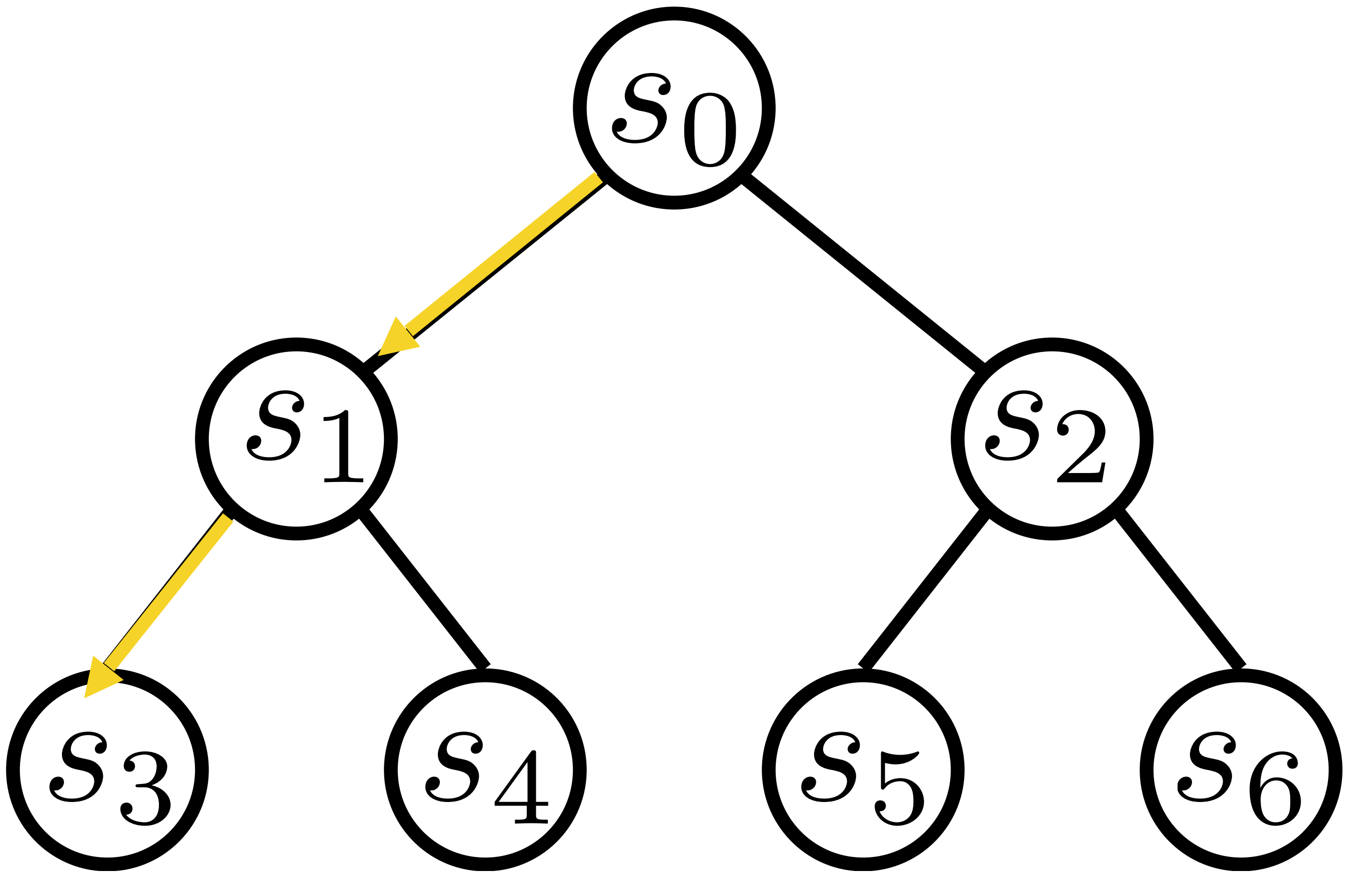
let us assume $\pi^e = \pi^*$, and we can query $Q^*(s, a)$



Halving: Eliminate half of the nodes at every iteration

[Sun et.al, 17,ICML]

In time logarithmic in #states, we know an optimal policy



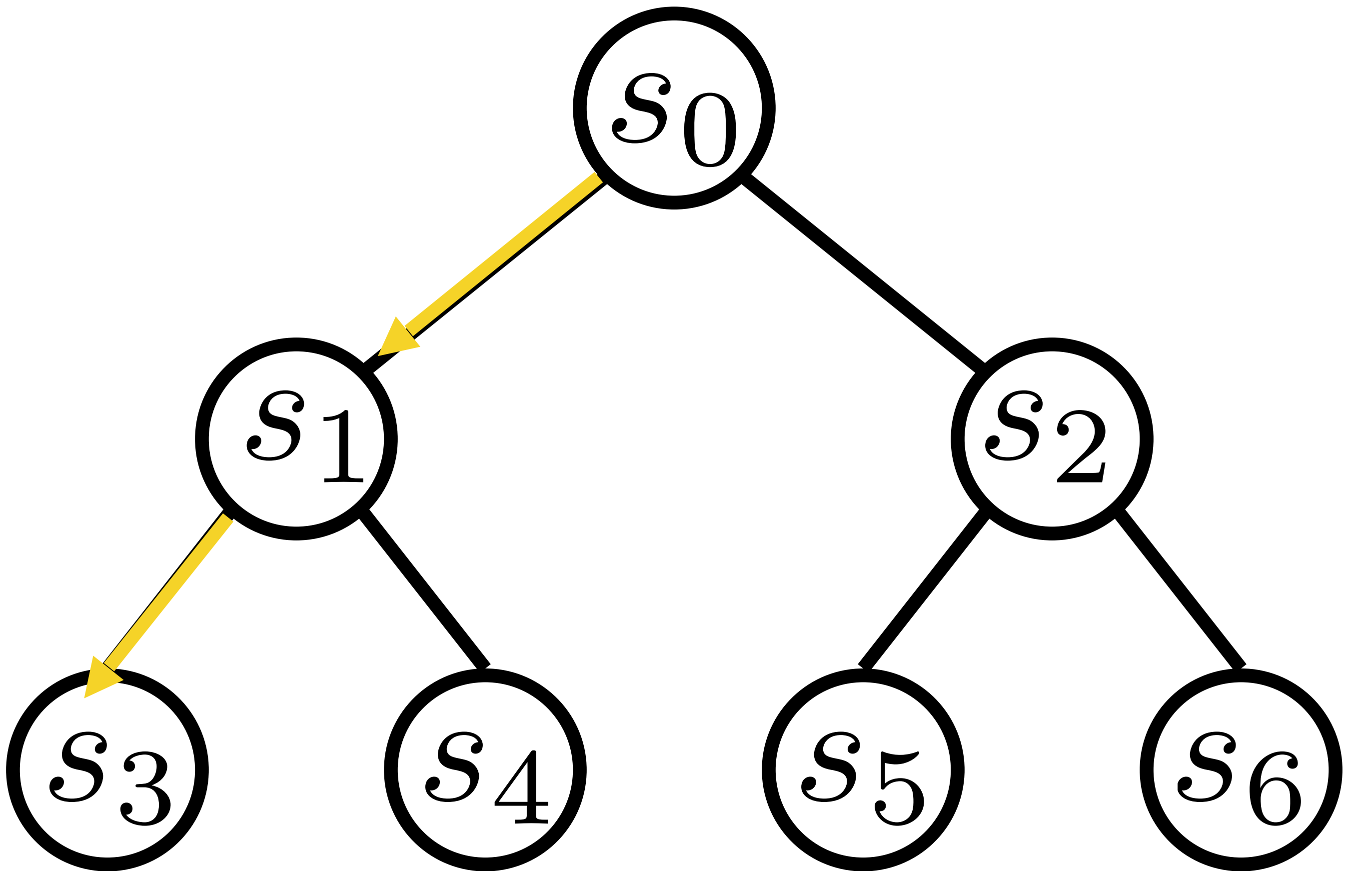
$$c_3 < c_4 < c_5 < c_6$$

$$\sum_{i=1}^{N} (J(\pi_i) - J(\pi^*)) \leq O(\log(S))$$

[Sun et.al, 17,ICML]

In time logarithmic in #states, we know an optimal policy

Now if we can only query **unbiased but noisy** Q^*

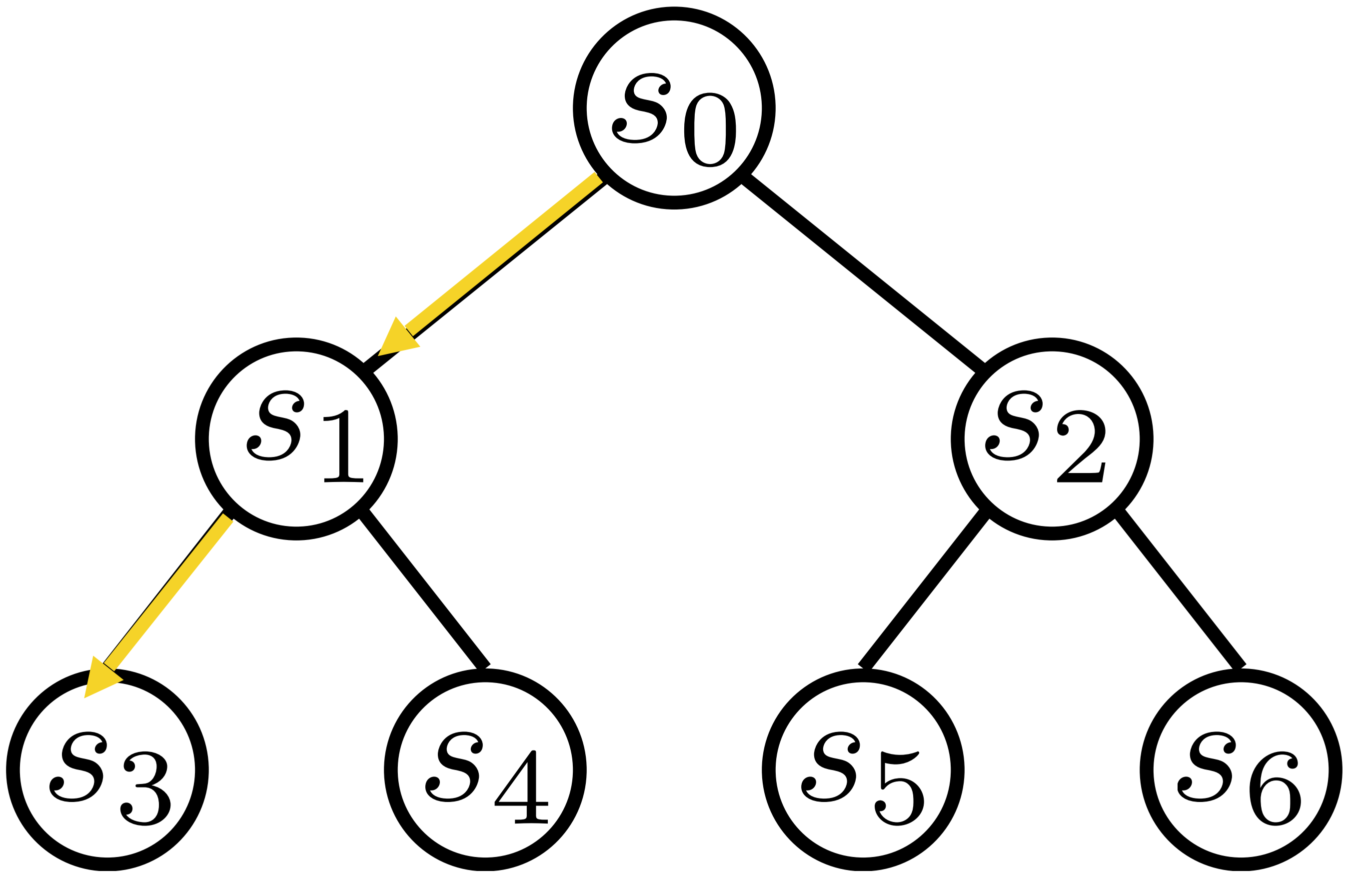


$c_3 < c_4 < c_5 < c_6$

$$\sum_{i=1}^{N}(J(\pi_i) - J(\pi^*)) \leq O\Big(\ln(S)(\sqrt{\ln(S)N} + \sqrt{\ln(2/\delta)N})\Big)$$

Poly-Log wrt S

But For Pure RL



$$c_3 < c_4 < c_5 < c_6$$

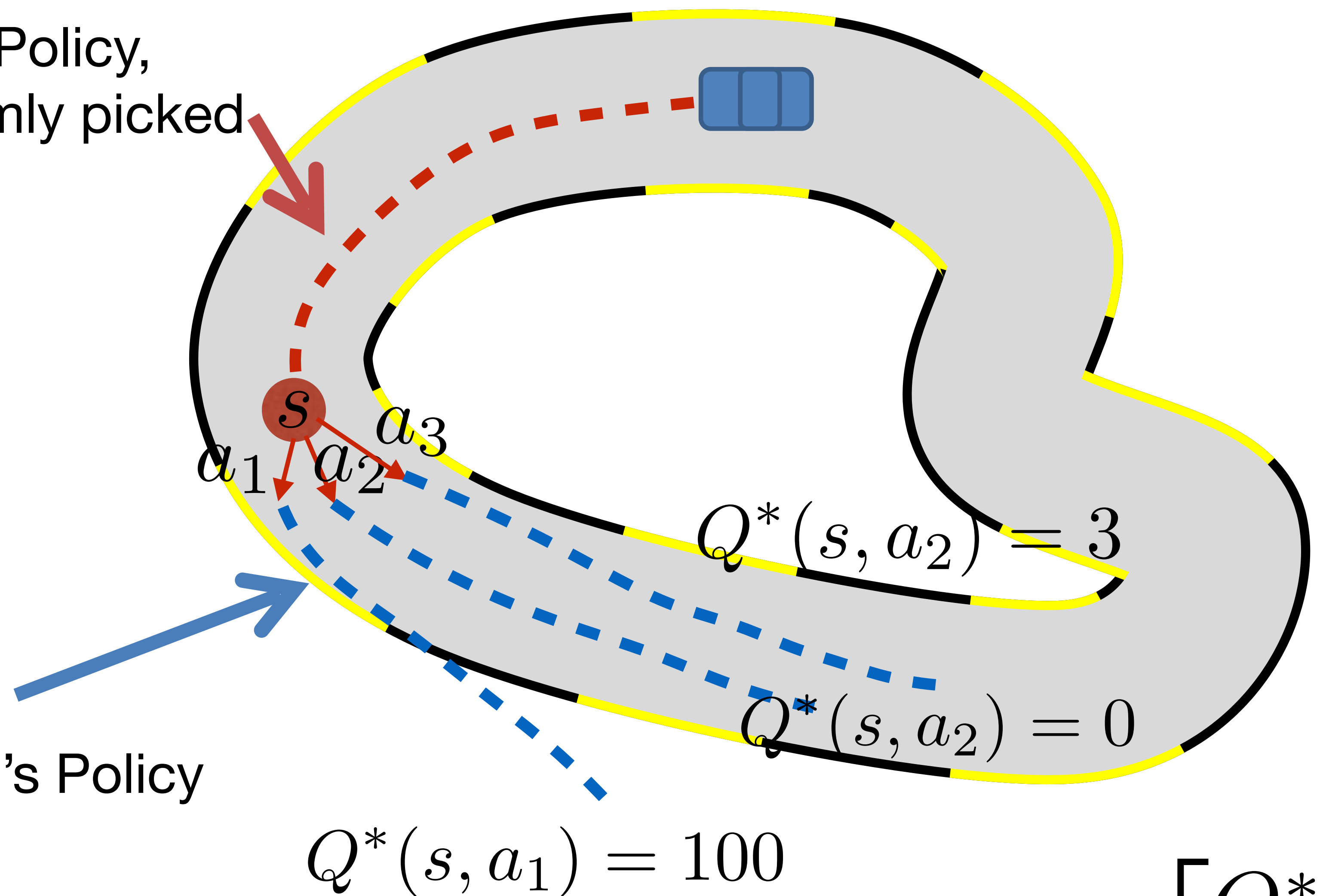
ANY RL ALGORITHM: $$\sum_{i=1}^{N} (J(\pi_i) - J(\pi^*)) \geq \Omega(\sqrt{SN})$$

Proof uses a reduction from Multi-Armed Bandit

[Ross & Bagnell, 14, arXiv]

Ex: AggreVaTe

Roll in Learned Policy,
Stop at a randomly picked time step



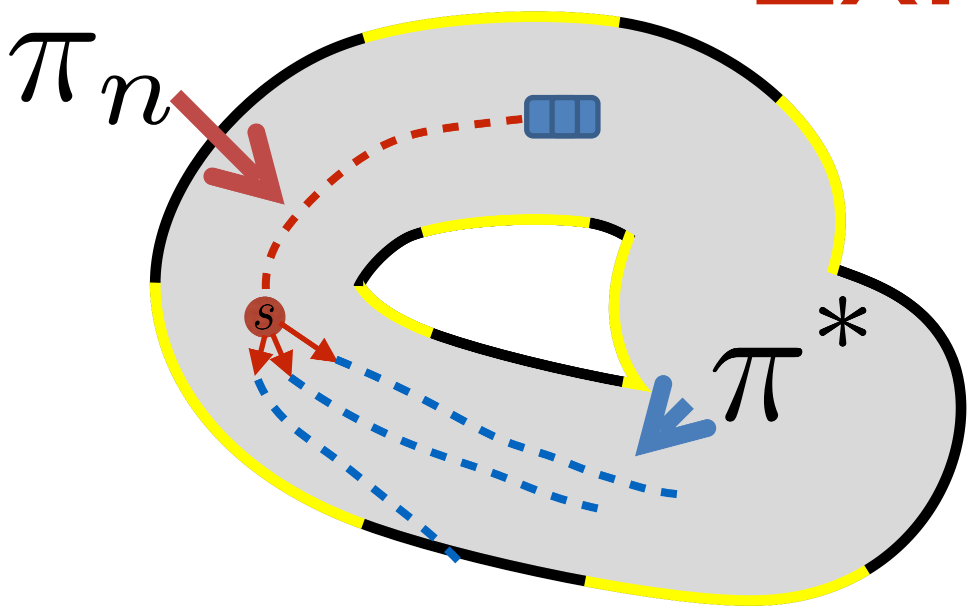
Roll out Expert's Policy

$$\left\{ s, \begin{bmatrix} Q^*(s, a_1) \\ Q^*(s, a_2) \\ Q^*(s, a_3) \end{bmatrix} \right\}_N$$

Cost-Sensitive classification dataset

Ex: AggreVaTe

[Ross & Bagnell, 14, arXiv]



$$\left\{ s, \begin{bmatrix} Q^*(s, a_1) \\ \dots \\ Q^*(s, a_A) \end{bmatrix} \right\}_N$$

Aggregate Dataset

$$\mathcal{D} + \left\{ s, \begin{bmatrix} Q^*(s, a_1) \\ \dots \\ Q^*(s, a_A) \end{bmatrix} \right\}_N$$

π_{n+1}

Cost-Sensitive Classifier

$$\pi_{n+1} = \arg\min_{\pi} \sum_{s \in \mathcal{D}} \sum_{a} \pi(a|s) Q^*(s, a)$$

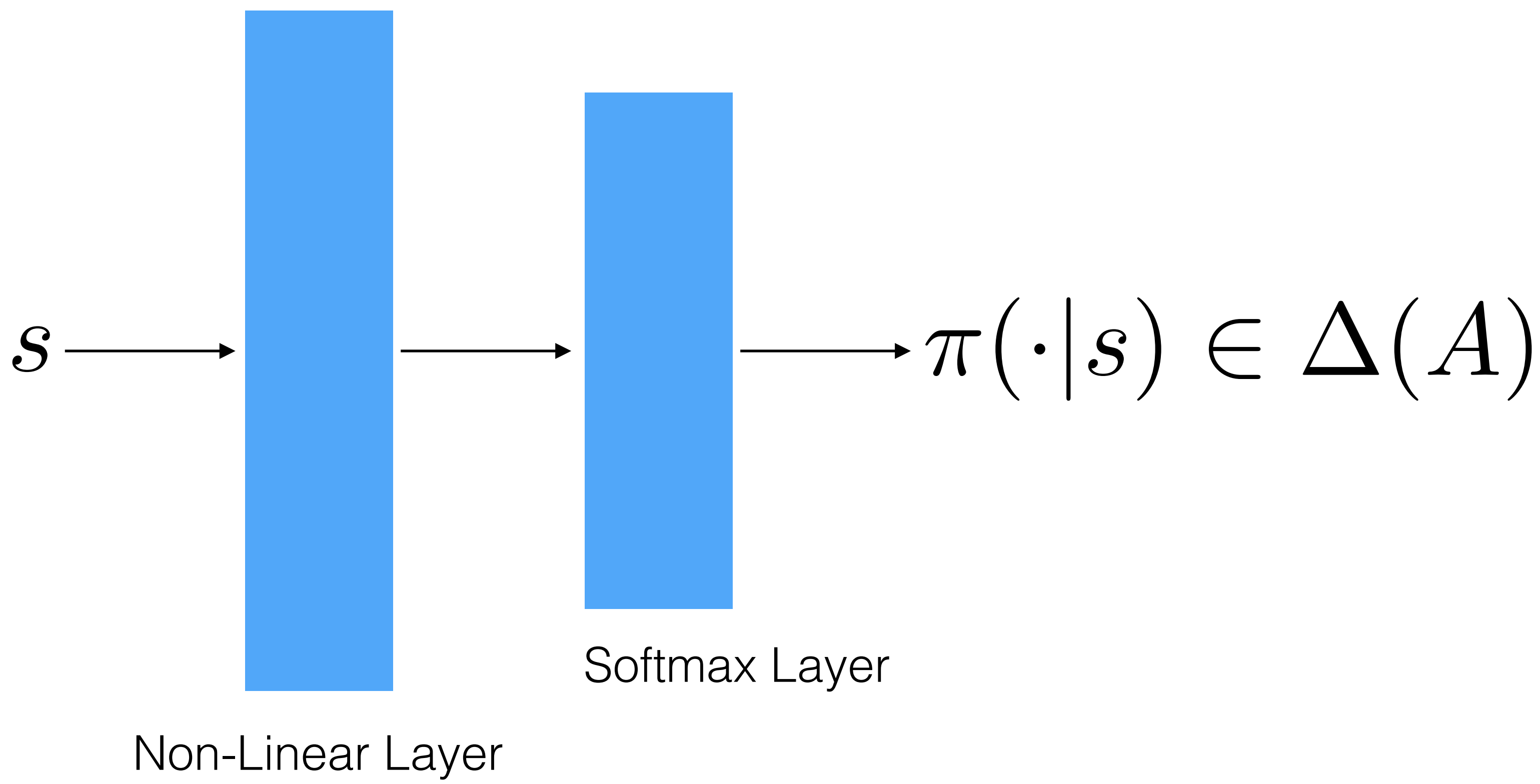
Simple incremental update?

Towards Differentiable AggreVaTe (AggreVaTe**D**)

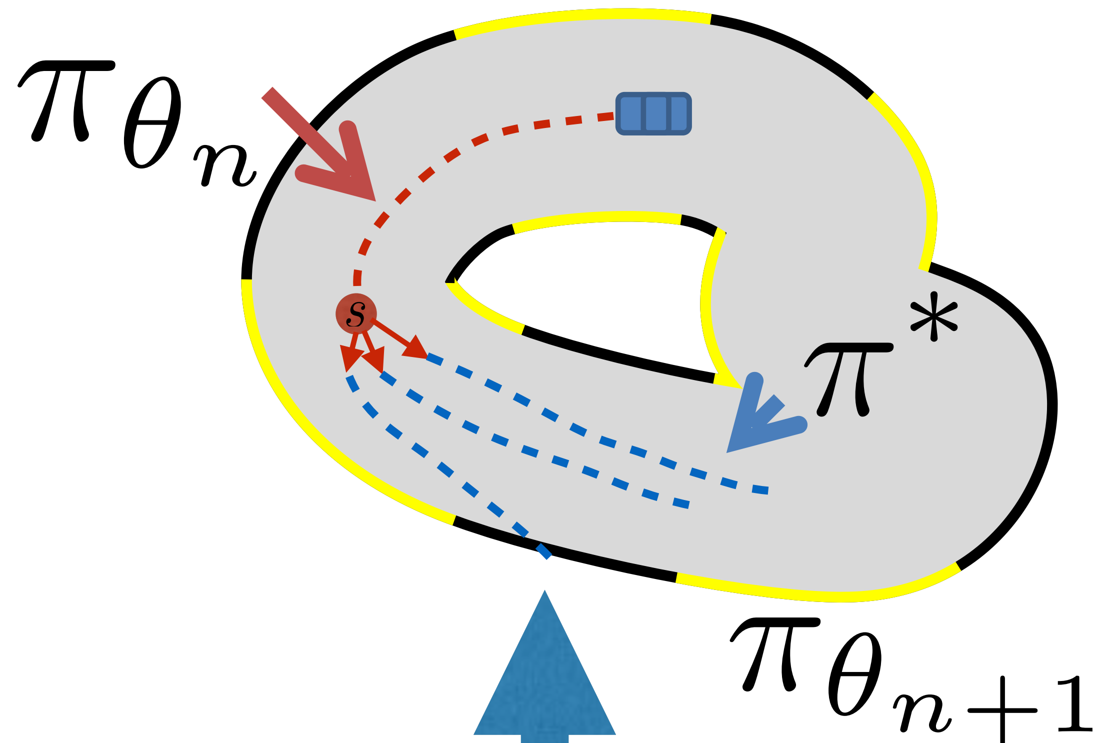
[Sun et.al, 17, ICML]

Stochastic parameterized policy:

$$\pi_\theta = \pi(\cdot|s;\theta)$$



Differentiable AggreVaTe (AggreVaTe**D**)



$$\left\{ s, \begin{bmatrix} Q^*(s, a_1) \\ \dots \\ Q^*(s, a_A) \end{bmatrix} \right\}_N$$

$$\ell_n(\theta) = \sum_s \sum_a \pi(a|s;\theta) Q^*(s,a)$$

Cost-Sensitive loss on the *new batch (no aggregation)*

$$\nabla_\theta \ell_n(\theta)|_{\theta_n}$$

AggreVaTeD-GD (Gradient Descent):

$$\theta_{n+1} = \theta_n - \mu \nabla_\theta \ell_n(\theta)|_{\theta=\theta_n}$$

AggreVaTeD-NG (Natural Gradient):

$$\theta_{n+1} = \theta_n - \eta_n I(\theta_n)^{-1} \nabla_{\theta_n} \ell_n(\theta_n)$$

[Sun et.al, 17, ICML]

Differentiable AggreVaTe (AggreVaTe**D**)

Discrete MDP, Tabular Policy:

AggreVaTeD-GD ⟷ AggreVaTe with Online Gradient Descent [Zinkevich, 03]

AggreVaTeD-NG ⟷ AggreVaTe with Weighted Majority [Littlestone & Warmuth, 03]

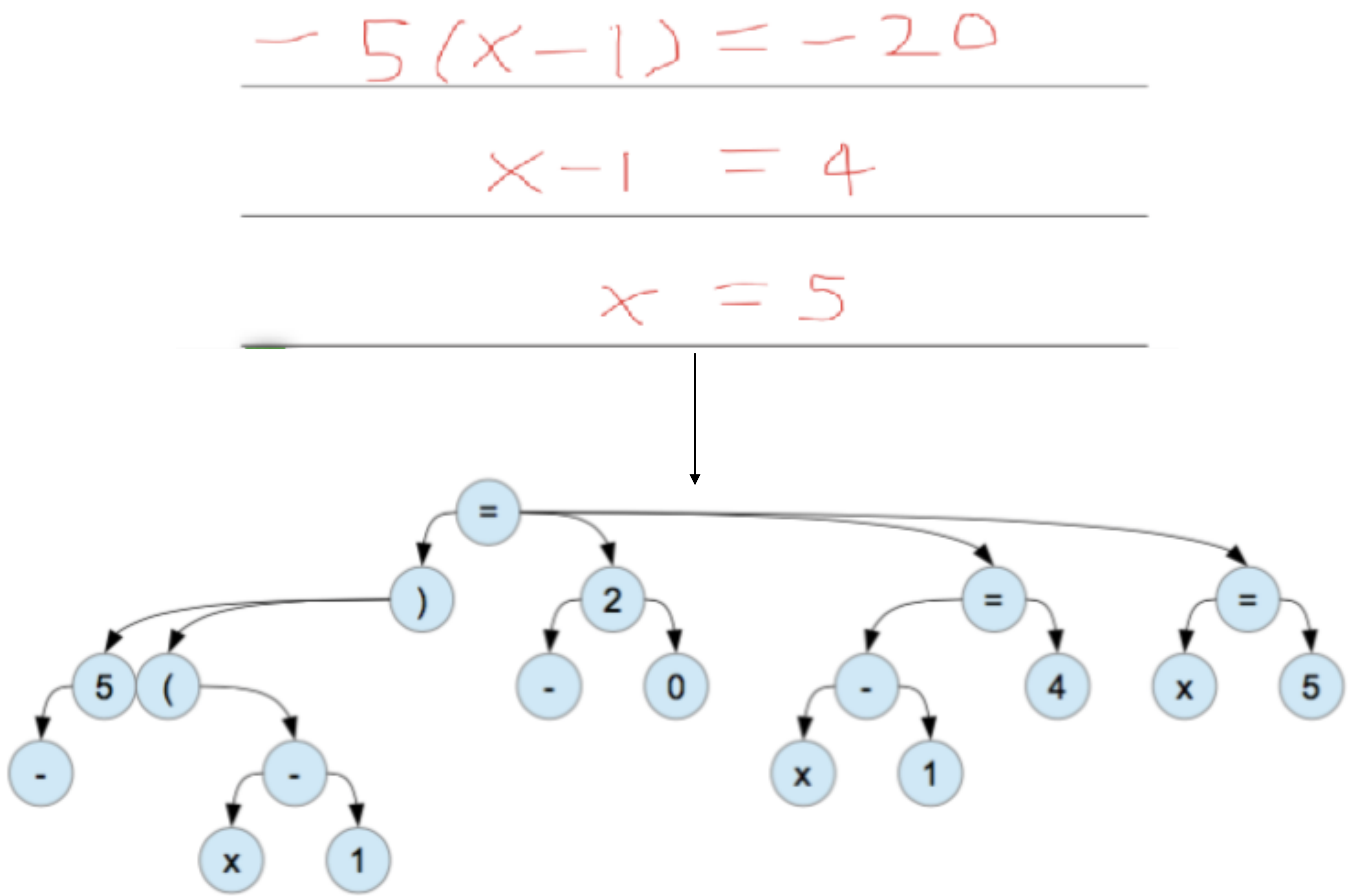
- *Practical* Algorithms
- AggreVaTe's theory as *Guidance*
- GD ensures *Convergence*

Strong Theoretical Guarantee

$$J(\hat{\pi}) \leq J(\pi^e)$$

Dependency Parsing

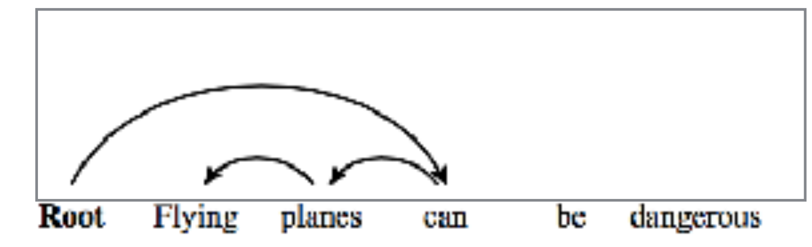
Dependency Parsing on Handwritten Algebra Data

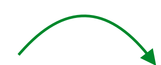


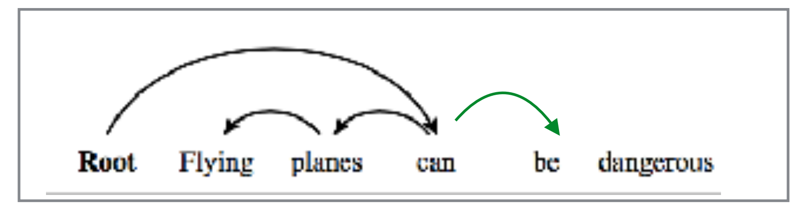
Dependency Parsing

Dependency Parsing <=> Sequential Decision Making

[Chang et.al 15, Duyck & Gordon 15]

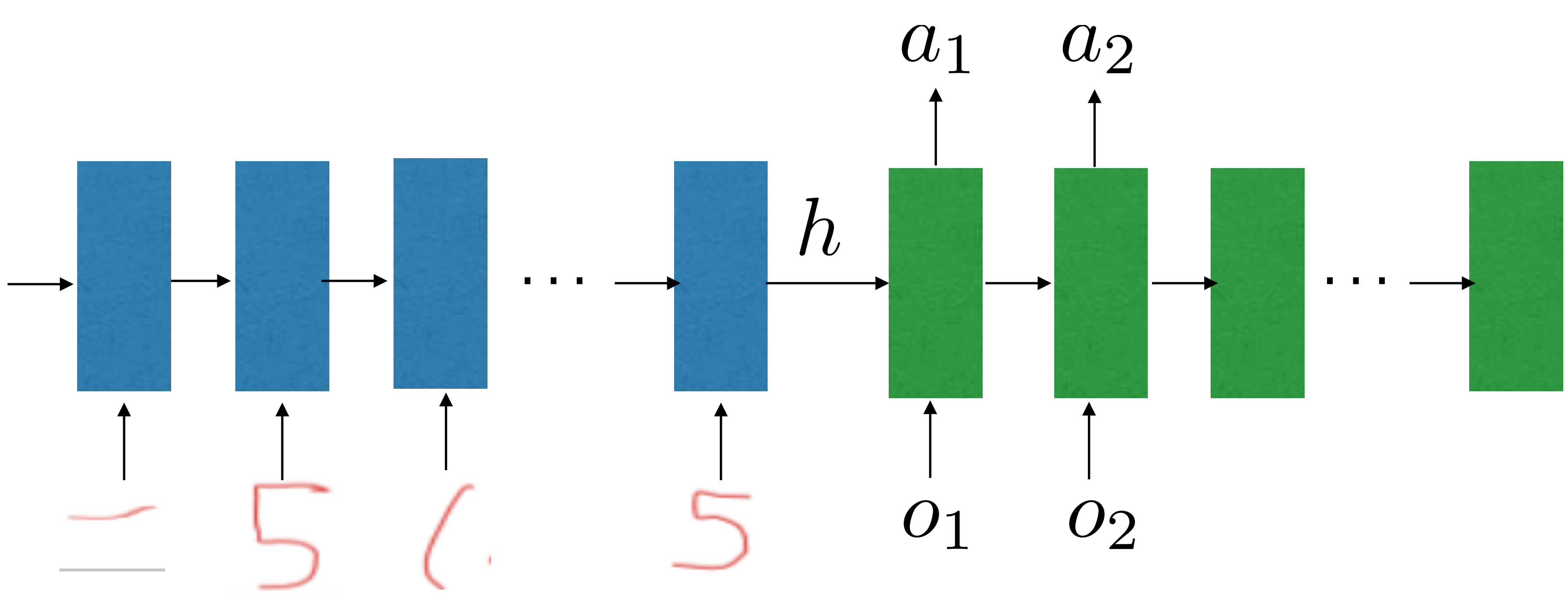




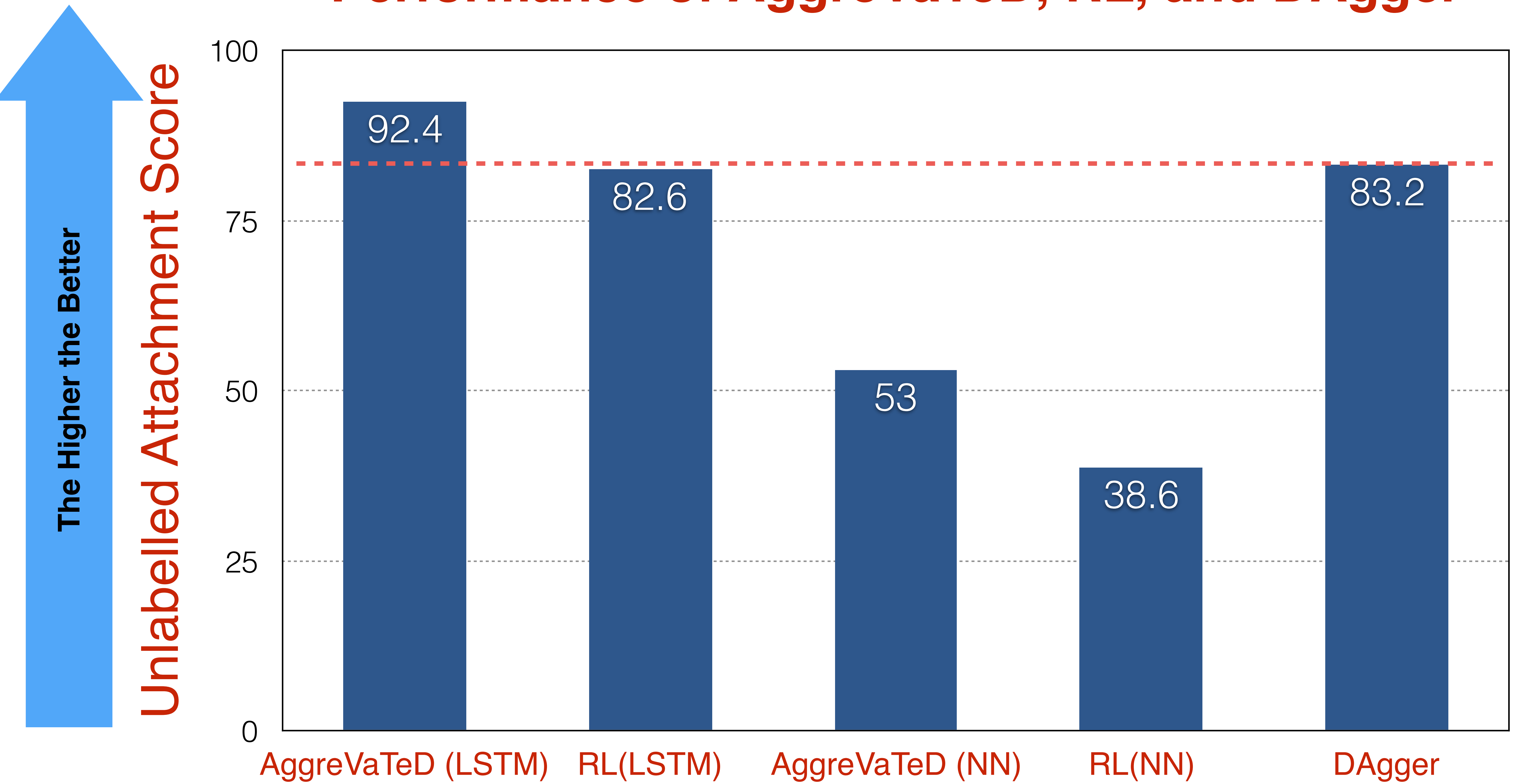


Partial constructed parse tree + Action (Arc) => New partial parse tree

$s_t \quad + \quad a_t \quad => \quad s_{t+1}$

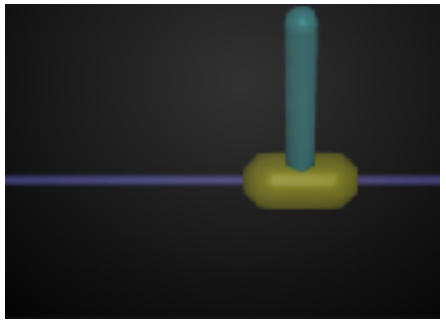


Performance of AggreVaTeD, RL, and DAgger

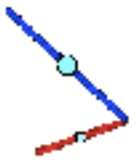


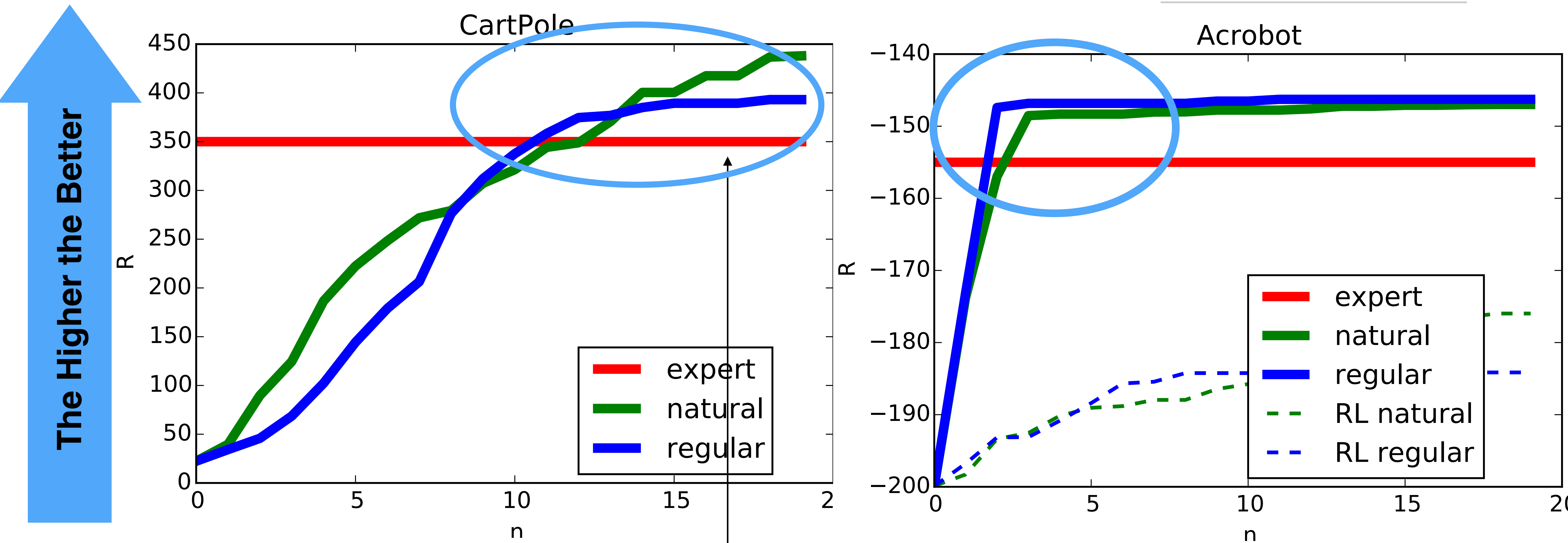
RL: Natural Policy Gradient [Kakade02, Bagnell 04]
DAgger result from Duyck & Gordon, 15

“Surprisingly”
It Can Outperform Expert



CartPole and Acrobot





R (y-axis): total reward

Experts