Towards Generalization and Efficiency in Reinforcement Learning

Wen Sun

Carnegie Mellon University

Joint work with Drew Bagnell, Geoff Gordon, Byron Boots, John Langford, Stephane Ross, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, Arun Venkatraman

Carnegie Mellon
THE ROBOTICS INSTITUTE

Goal:

Design Algorithms that have
Generalization & Sample Efficiency
in learning to make decisions
in complex environments

My Research

1. Expert Demonstration

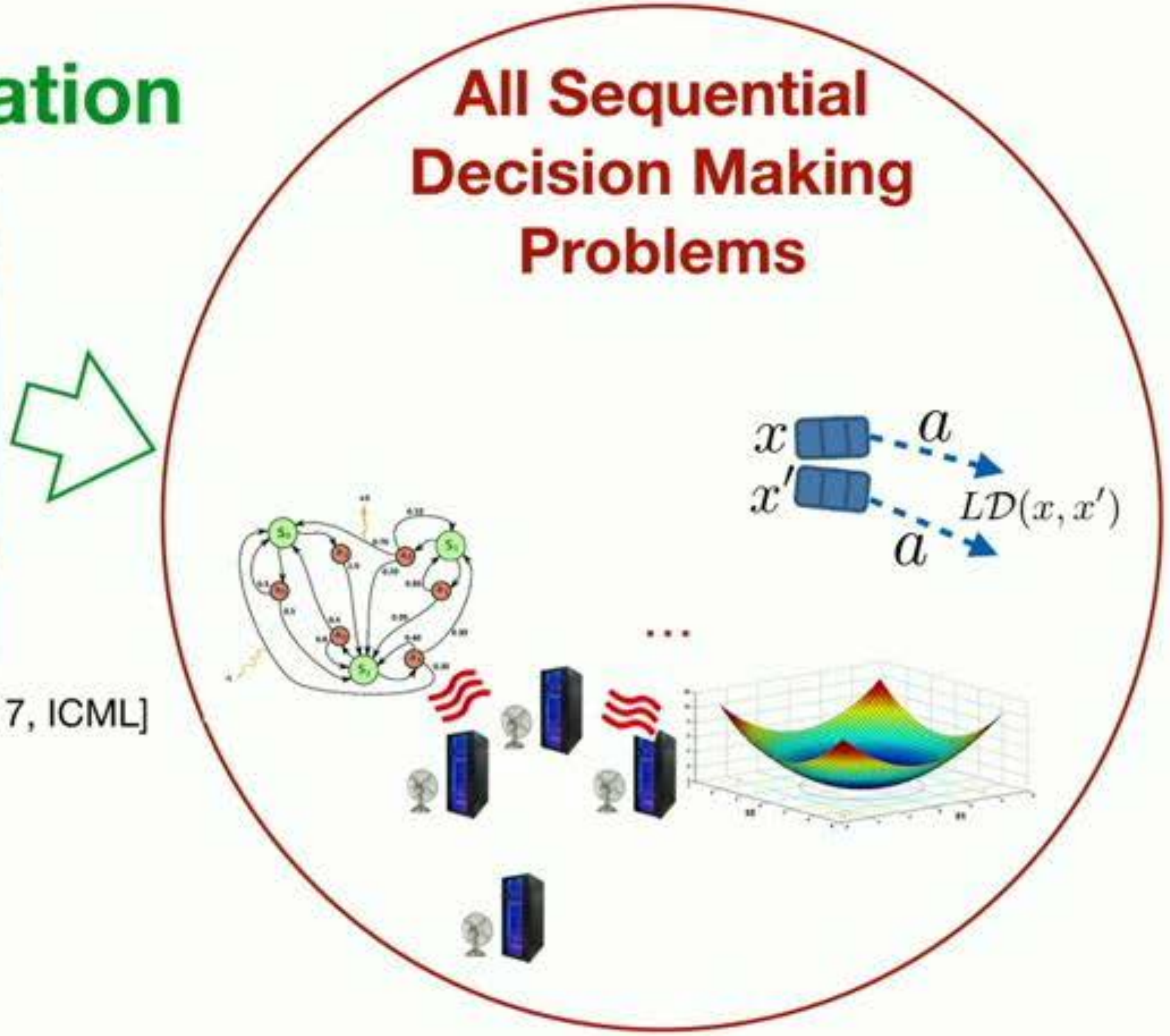
[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]
[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]

All Sequential Decision Making Problems

My Research

1. Expert Demonstration

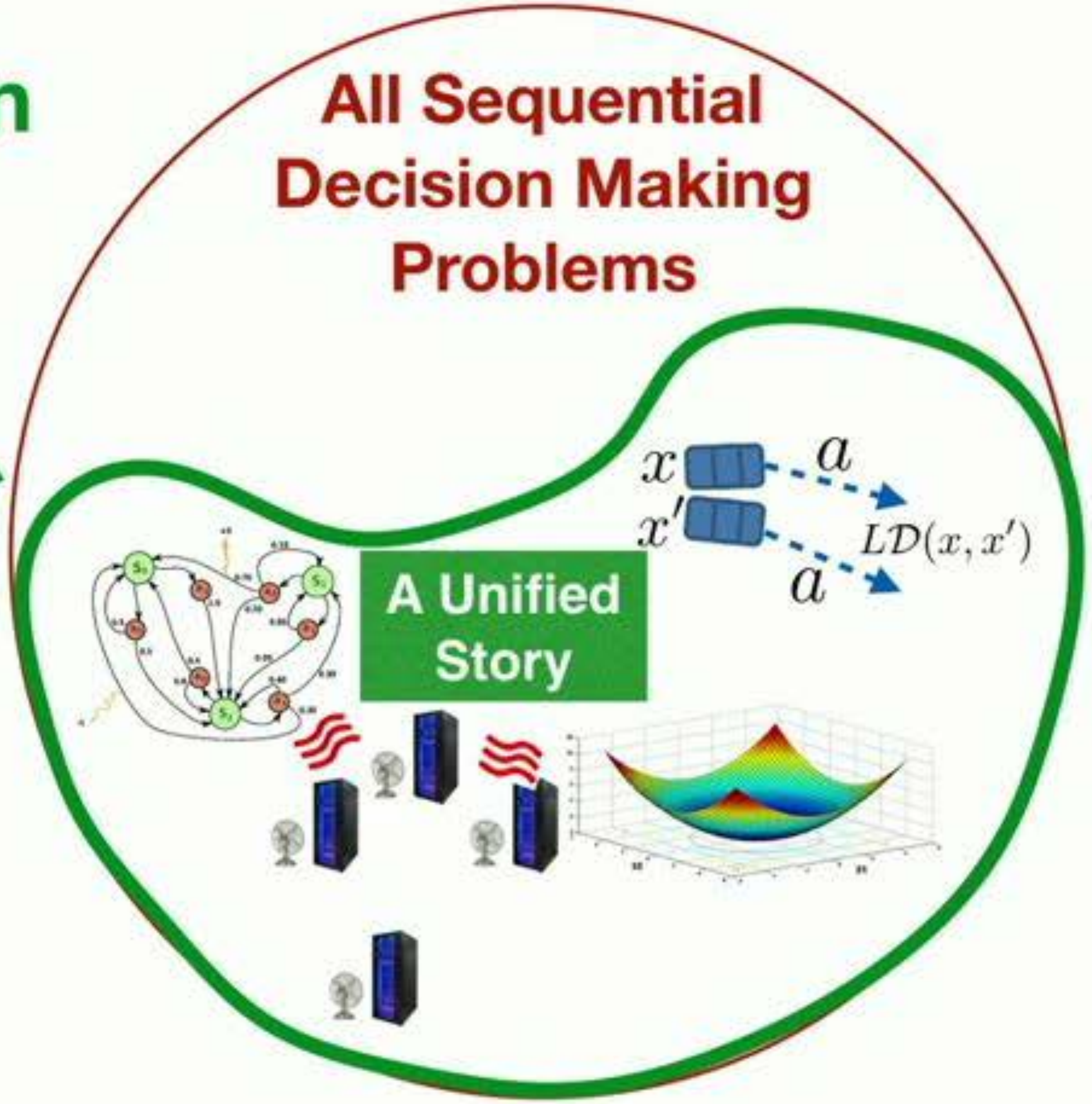
[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]
[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]



My Research

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]
[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]



2. Exploiting Structures

[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Supervised Learning VS Sequential Decision Making

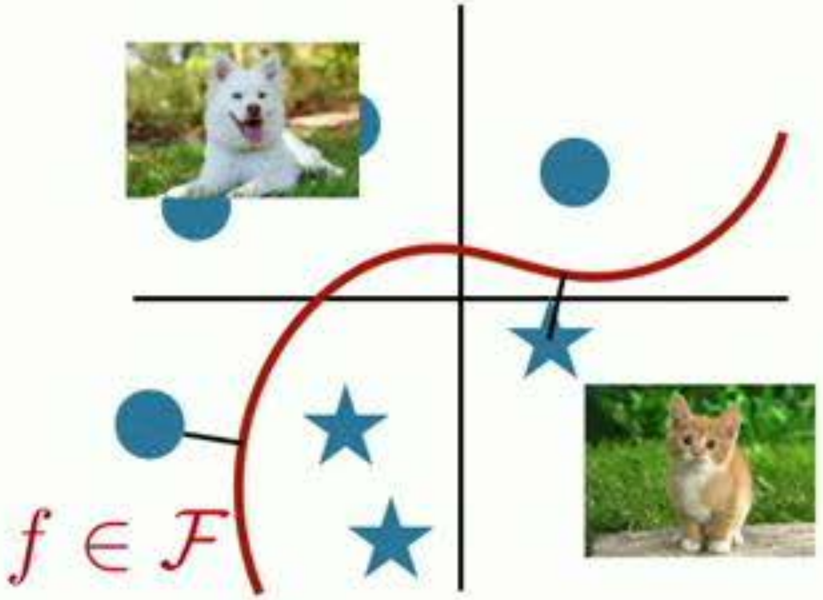
Given i.i.d examples at training:

( ,cat) ( ,cat) ( ,dog)

Supervised Learning VS Sequential Decision Making

Given i.i.d examples at training:

( ,cat) ( ,cat) ( ,dog)



Passive:

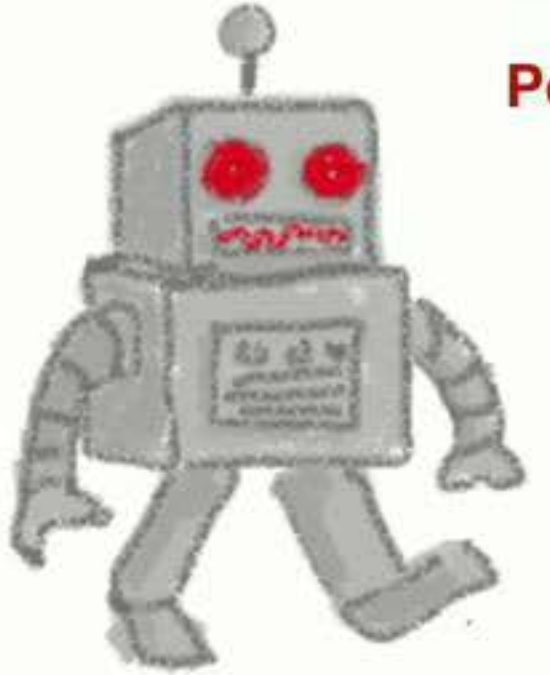
Active: Decisions → Data Distribution

[Ross&Bagnell, 11, AISTATS]

Reinforcement Learning

Markov Decision Process

Agent



$$\pi(x) \rightarrow a$$

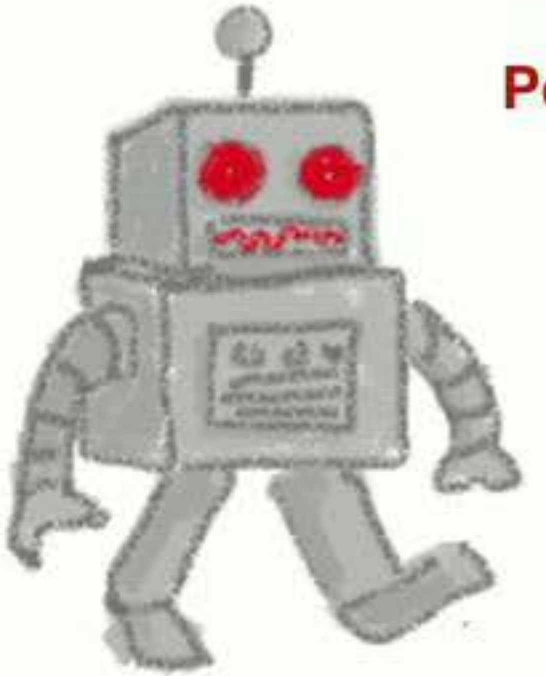
Policy: determine **action** based on **state**

Environment

Reinforcement Learning

Markov Decision Process

Agent



$$\pi(x) \to a$$

Policy: determine **action** based on **state**

Environment

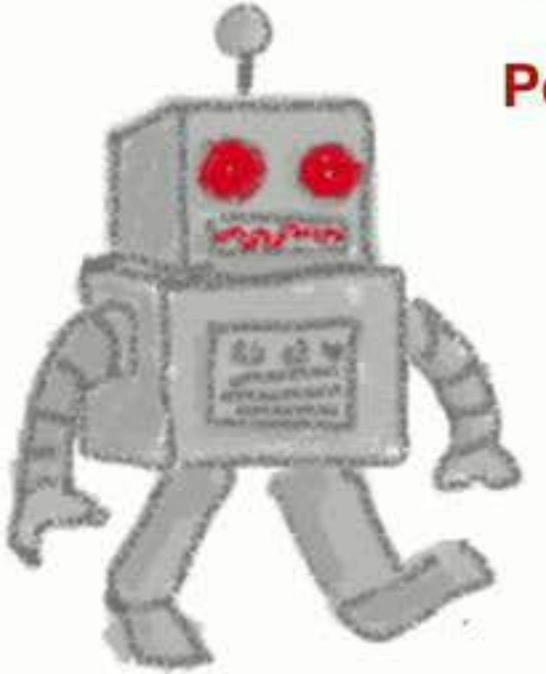
Send **reward** and **next state** from a Markovian transition dynamics

$$r(x, a), \quad x' \sim P(\cdot|x, a)$$

Reinforcement Learning

Markov Decision Process

Agent



$$\pi(x) \rightarrow a$$

Policy: determine **action** based on **state**

H Steps

Send **reward** and **next state** from a Markovian transition dynamics

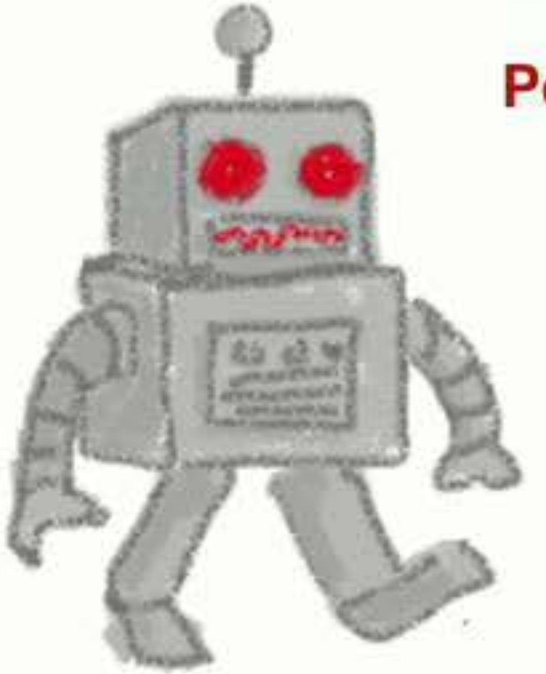
$$r(x,a), \quad x' \sim P(\cdot|x,a)$$

Environment

Reinforcement Learning

Markov Decision Process

Agent



Environment

$$\pi(x) \rightarrow a$$

Policy: determine **action** based on **state**

H Steps

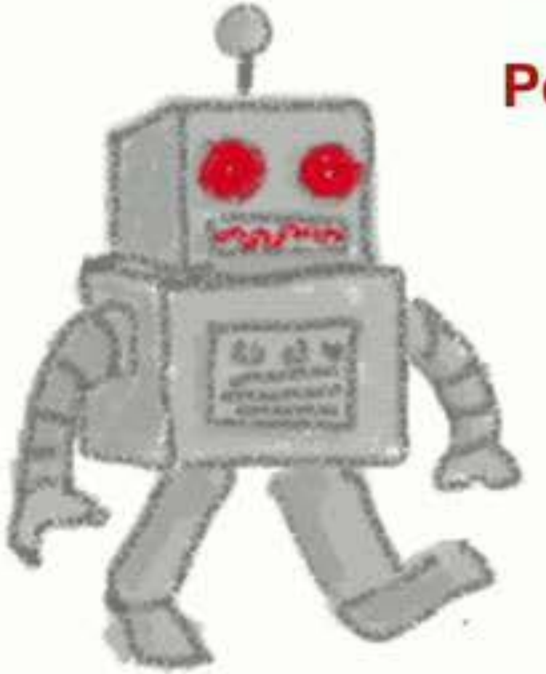
Send **reward** and **next state** from a Markovian transition dynamics

$$r(x, a), \quad x' \sim P(\cdot|x, a)$$

Reinforcement Learning

Markov Decision Process

Agent



$$\pi(x) \rightarrow a$$

Policy: determine **action** based on **state**

H Steps

Send **reward** and **next state** from a Markovian transition dynamics

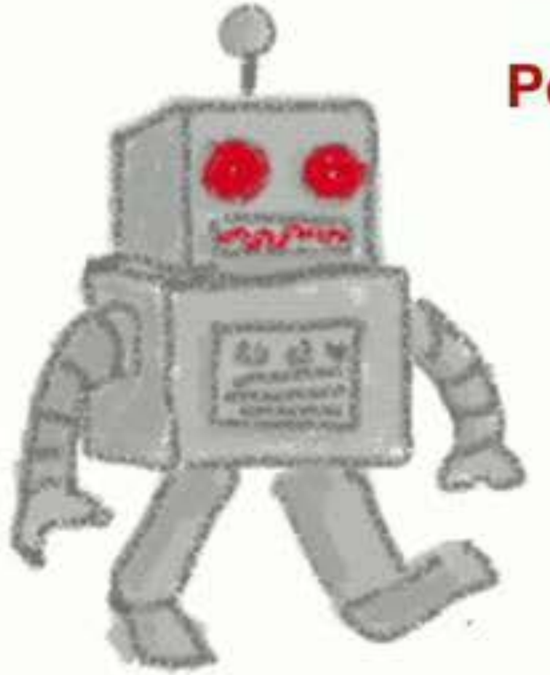
$$r(x, a), \quad x' \sim P(\cdot|x, a)$$

Environment

Reinforcement Learning

Markov Decision Process

Agent



Environment

$$\pi(x) \to a$$

Policy: determine **action** based on **state**

H Steps

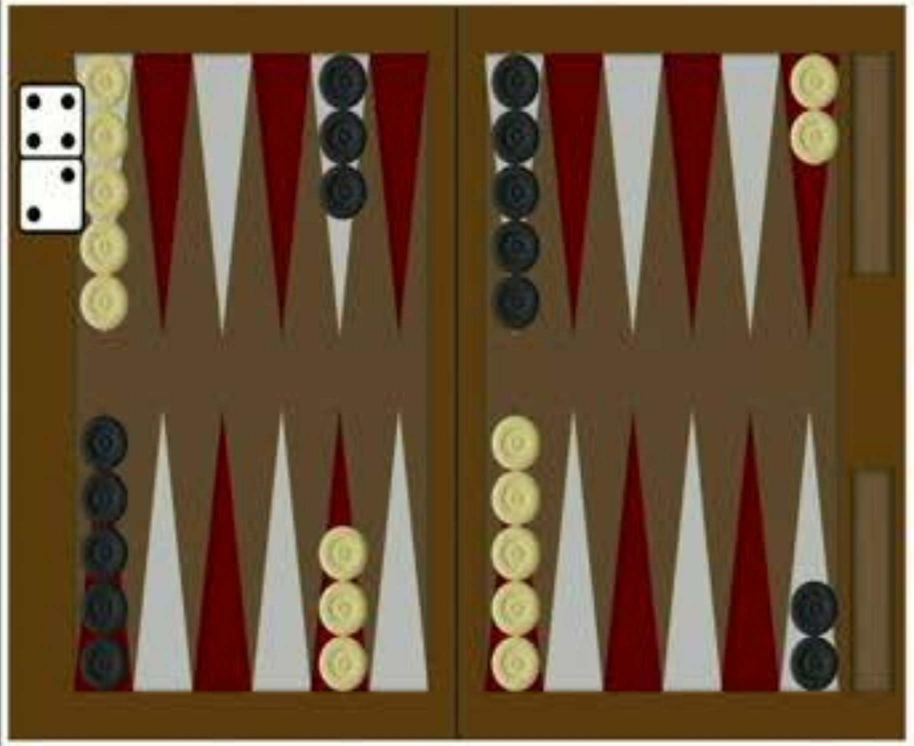
Send **reward** and **next state** from a Markovian transition dynamics

$$r(x, a), \quad x' \sim P(\cdot | x, a)$$

Maximize expected total reward:

$$J(\pi) = \mathbb{E}[r_1 + r_2 + \cdots + r_H | \pi]$$

Progress of RL in Practice



TD GAMMON [Tesauro 95]

[AlphaZero, Silver et.al, 17]

[OpenAI Five, 18]

Progress of RL in Practice

*OpenAI Five plays 180 years worth of games against itself every day…..**running on 256 GPUs and 128,000 CPU cores***

—*Open AI Five Blog*

[OpenAI Five]

Progress of RL in Practice

[OpenAI Five]

Inefficient Exploration

Random Trial and error via
massive simulation
(i.e., **128,000** CPUs)

Inefficient Exploration

Random Trial and error via
massive simulation
(i.e., **128,000** CPUs)

$\neq$

Inefficient Exploration

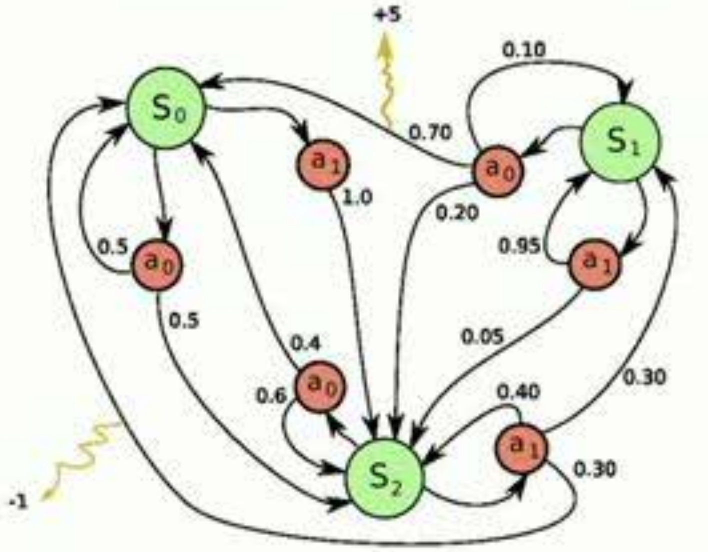
Random Trial and error via
massive simulation
(i.e., **128,000** CPUs)

$\neq$

Sample Efficiency

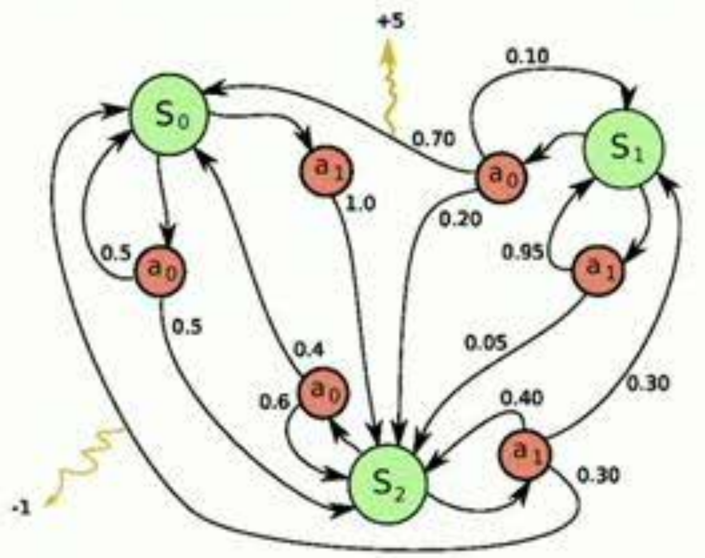
Progress of RL in Theory

Sample Efficiency in Small Discrete MDPs



Progress of RL in Theory

Sample Efficiency in Small Discrete MDPs



Sample Complexity:

To achieve ϵ near-optimal policy, need at most

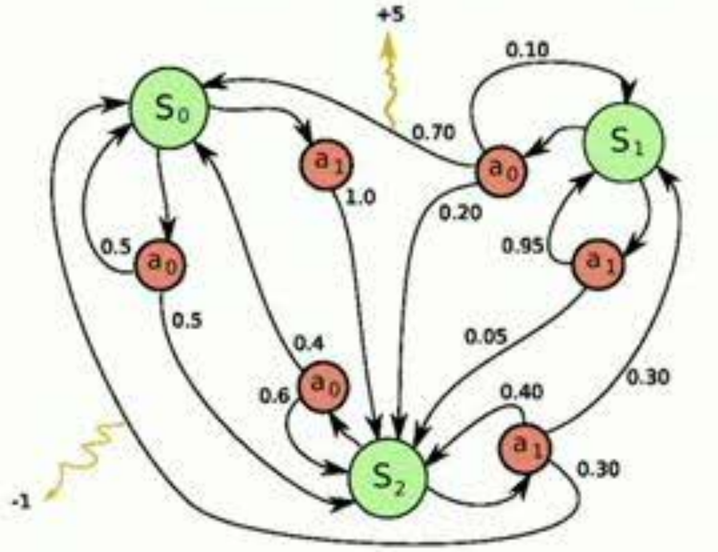
$$\text{poly}(\#\text{ of states}, \#\text{ of actions}, \text{Horizon}, 1/\epsilon)$$

many interactions

[e.g., Kearns & Singh 02, Dann & Brunskill, 15, Azar et.al, 17]

Progress of RL in Theory

Sample Efficiency in Small Discrete MDPs



Sample Complexity:

To achieve ϵ near-optimal policy, need at most

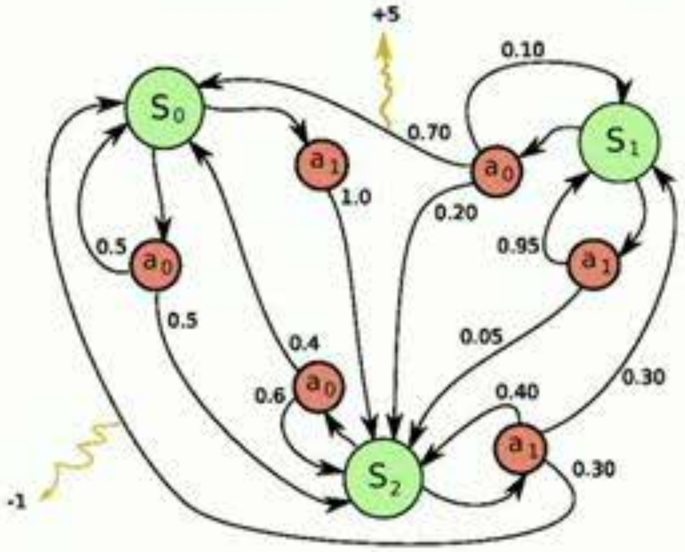
$\boxed{\text{poly}(\#\text{ of states,}}\ \#\text{ of actions, Horizon}, 1/\epsilon)$

many interactions

[e.g., Kearns & Singh 02, Dann & Brunskill, 15, Azar et.al, 17]

Progress of RL in Theory

Sample Efficiency in Small Discrete MDPs



Sample Complexity:

To achieve ϵ near-optimal policy, need at most

$$\boxed{\text{poly}(\#\text{ of states},}\ \#\text{ of actions, Horizon}, 1/\epsilon)$$

many interactions

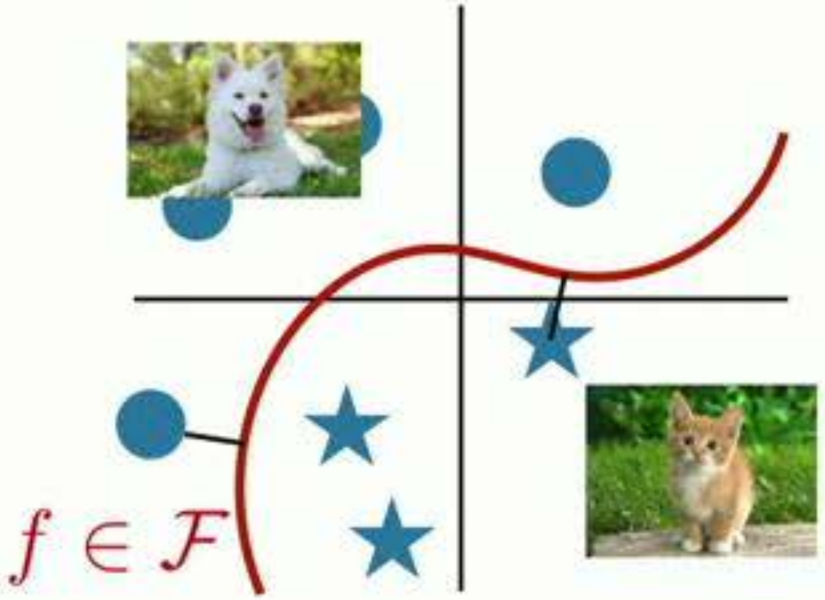
[e.g., Kearns & Singh 02, Dann & Brunskill, 15, Azar et.al, 17]

$\neq$

Large-Scale Decision Making Problems

What We Understand:
Supervised Learning

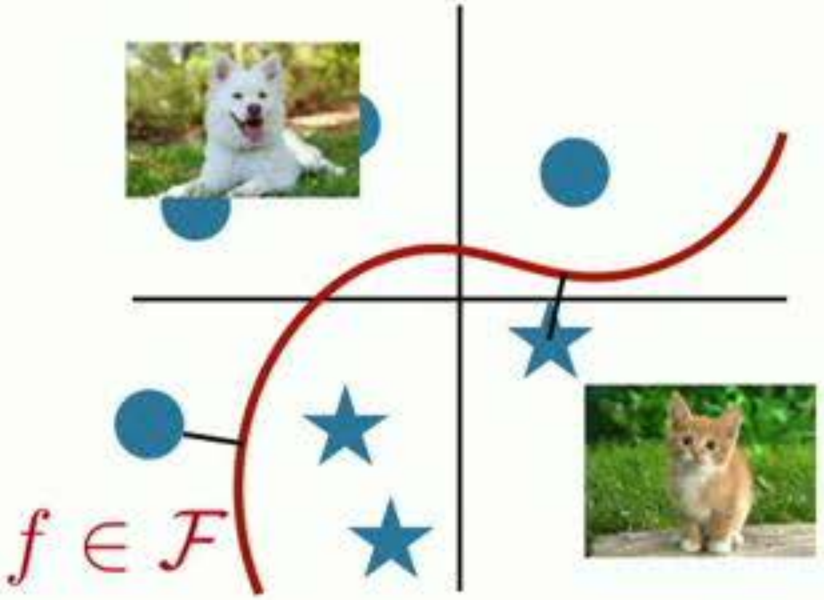
[ImageNet]



Polynomial Dependency ~~of #~~ **of unique images**

What We Understand:
Supervised Learning

[ImageNet]

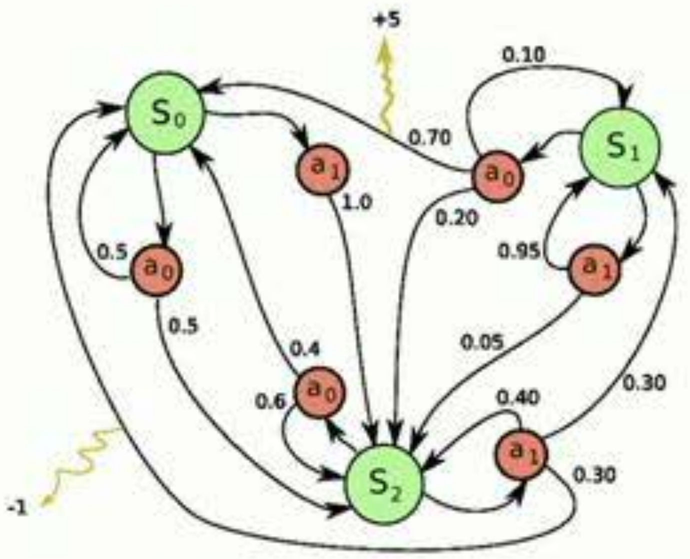


Polynomial Dependency of **# of unique images**

Generalization via Function Approximation

What We Want:
Generalization in Large-Scale MDPs

Sample Efficiency



Sample Complexity:

To achieve ϵ near-optimal policy, we need at most

$$\mathrm{poly}(\#\text{ of states}, \#\text{ of actions}, \text{Horizon}, 1/\epsilon)$$

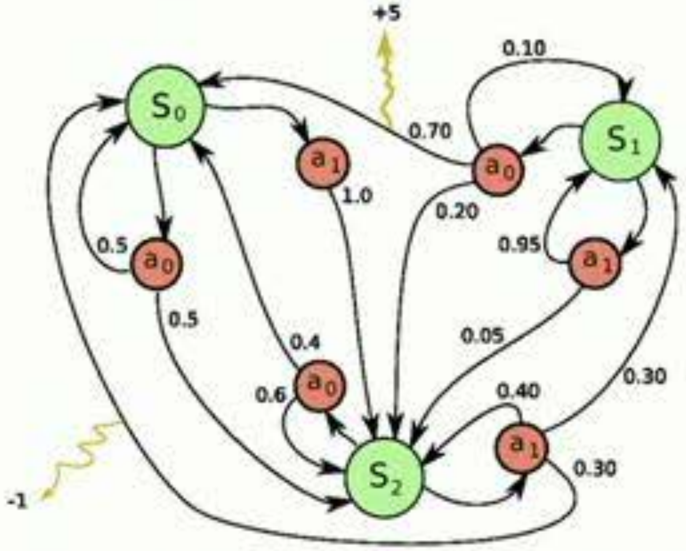
many interactions

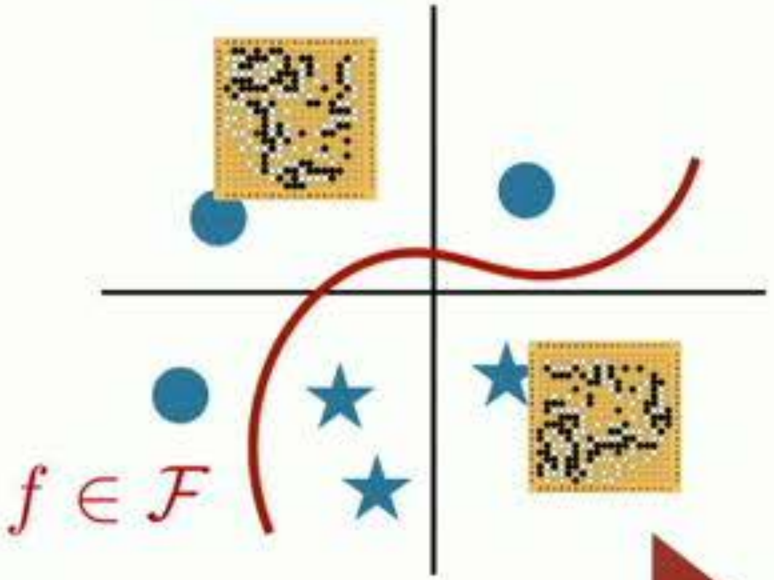
[e.g., Kearns & Singh 02, Dann & Brunskill, 15,Azar et.al, 17]

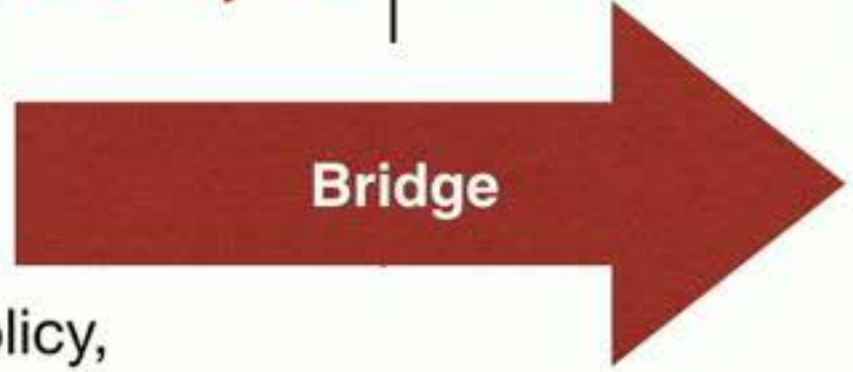
$\neq$

What We Want:
Generalization in Large-Scale MDPs

Sample Efficiency







Sample Complexity:

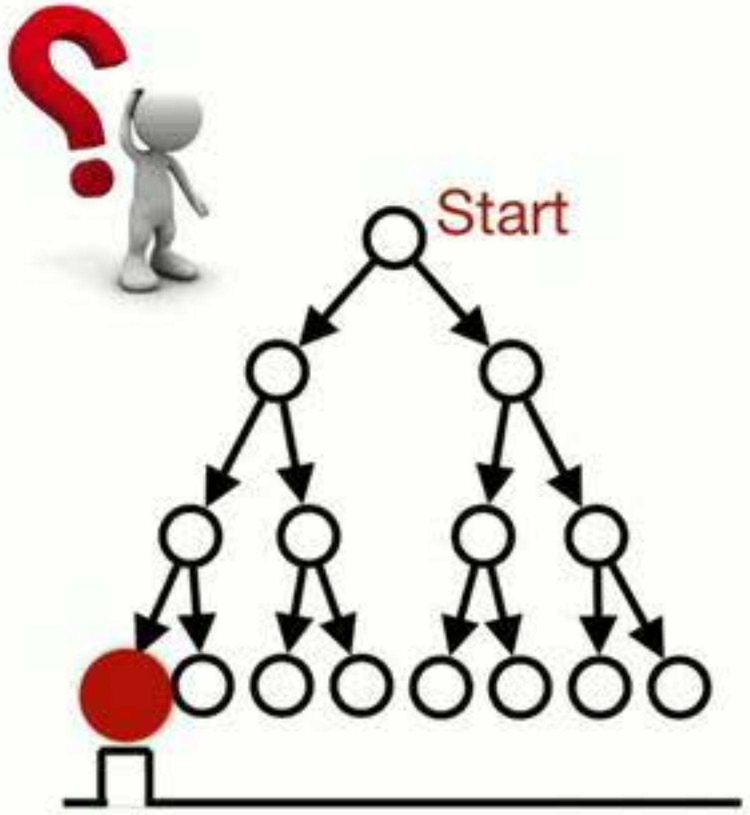
To achieve ϵ near-optimal policy, we need at most

$$\boxed{\text{poly}(\#\text{ of states},}\ \#\text{ of actions, Horizon}, 1/\epsilon)$$

many interactions

[e.g., Kearns & Singh 02, Dann & Brunskill, 15,Azar et.al, 17]

BUT…



Reward only at one leaf

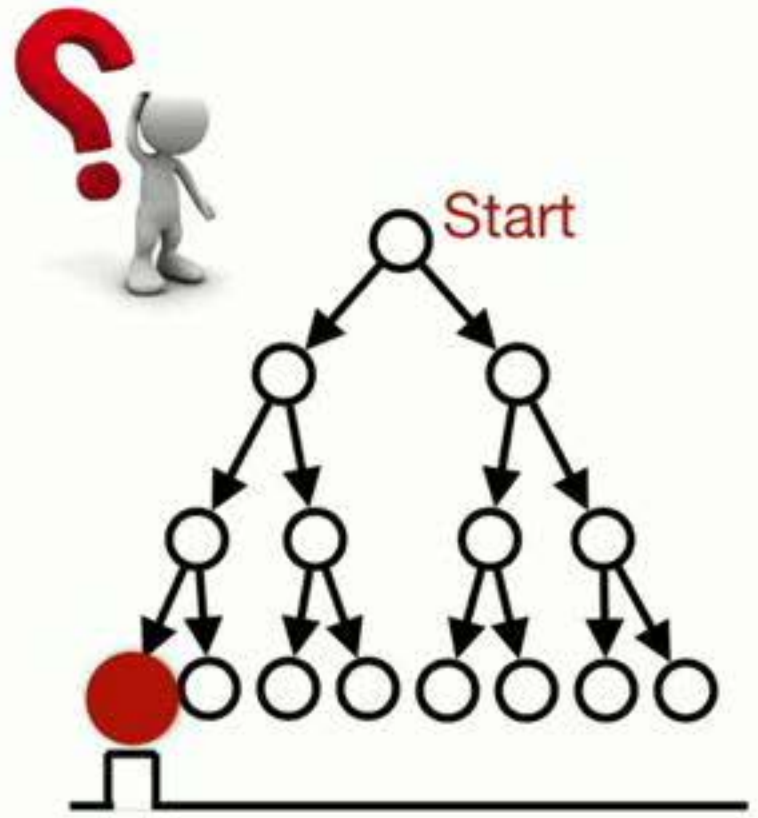
[e.g., Krishnamurthy et.al 16, Jiang et.al 17]

Needle in a haystack

Discrete MDPs

H: horizon, S: # of states, A: # of actions

BUT...



Reward only at one leaf

[e.g., Krishnamurthy et.al 16, Jiang et.al 17]

Needle in a haystack

Discrete MDPs

H: horizon, S: # of states, A: # of actions

# of Interactions with environment > $\Omega(S)$

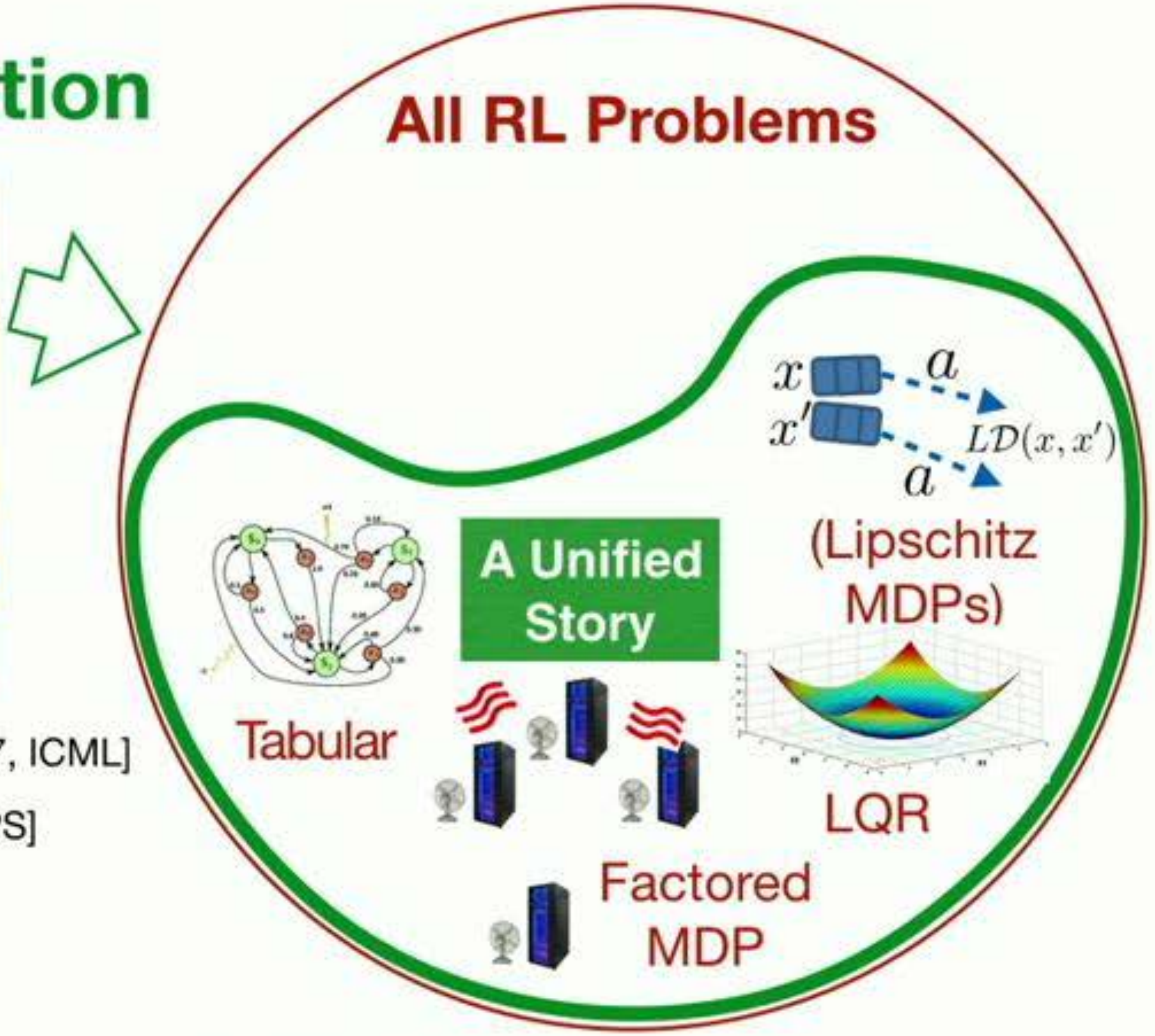
[e.g.,Dann & Brunskill, 15]

Generalization & Sample Efficiency via…

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]

[**Sun**,, Gordon, Boots, Bagnell, 18, NeurIPS]



2. Exploiting Structures

[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

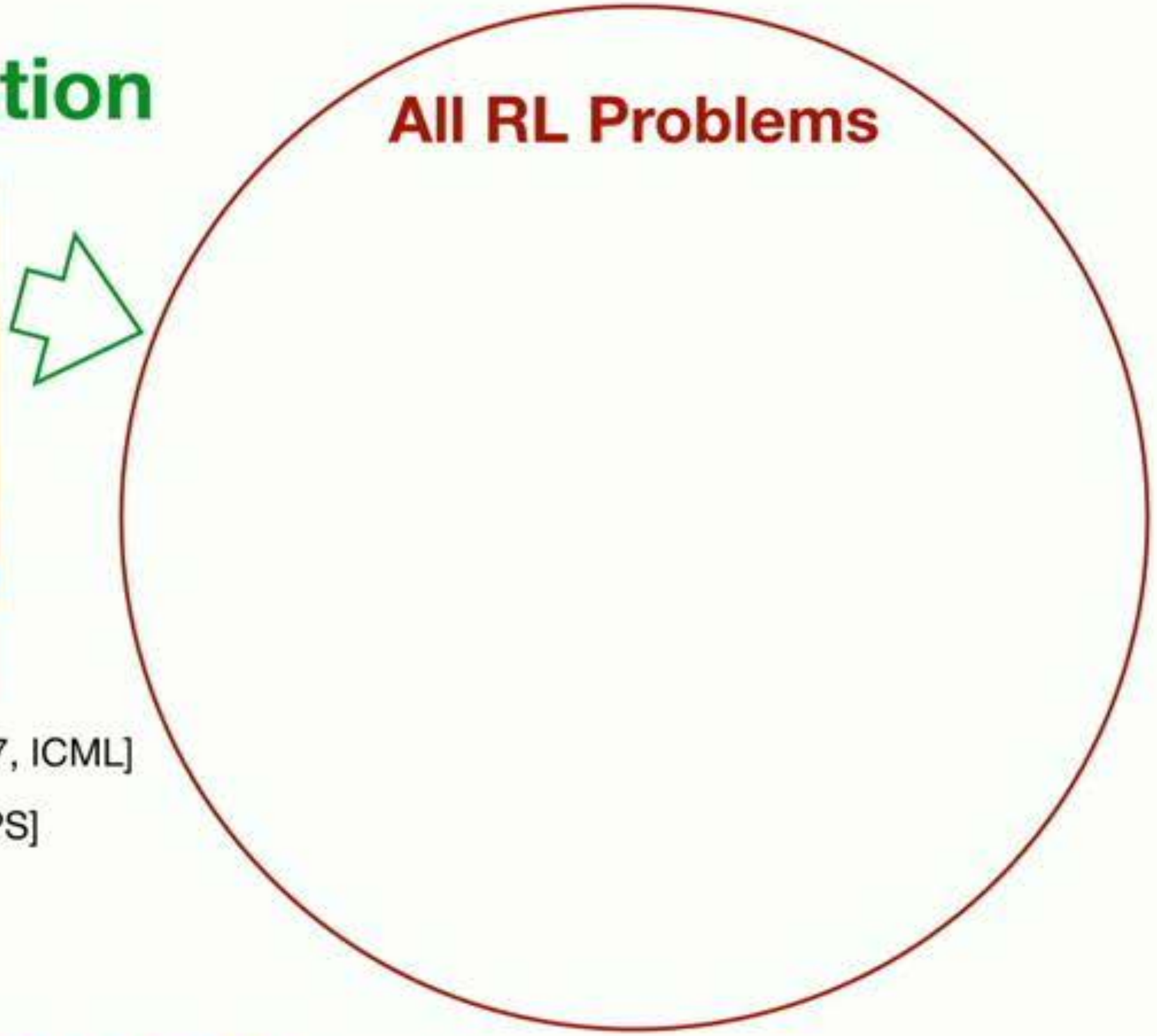
Generalization & Sample Efficiency via…

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]

[**Sun**,, Gordon, Boots, Bagnell, 18, NeurIPS]

- **Why** IL (i.e., IL VS RL)
- **How** to reduce RL to **Supervised Learning**
- **Generalize** from **Local** Experts



Imitation Learning

Maps states to actions

- SVM
- Gaussian Process
- Deep Networks

Apprenticeship Learning [Abbeel & Ng 05, Syed & Schapire 08]

Inverse Optimal Control [Ziebart & Bagnell, 10]

Interactive Imitation Learning [Ross& Bangell, 11; Chang et.al., 15]

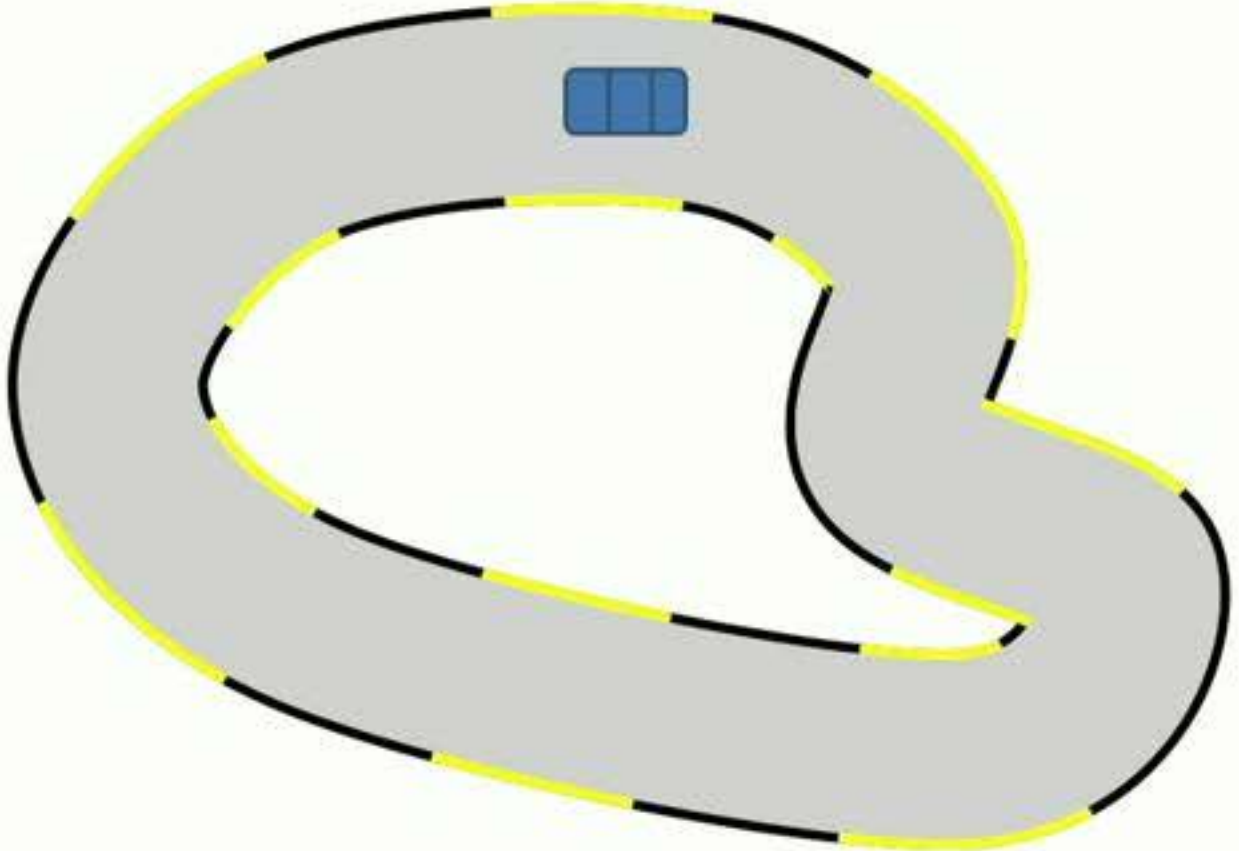
Generative Adversarial Imitation Learning [Ho & Ermon 16]

Interactive Imitation Learning w/ Reward

A global expert is available during training

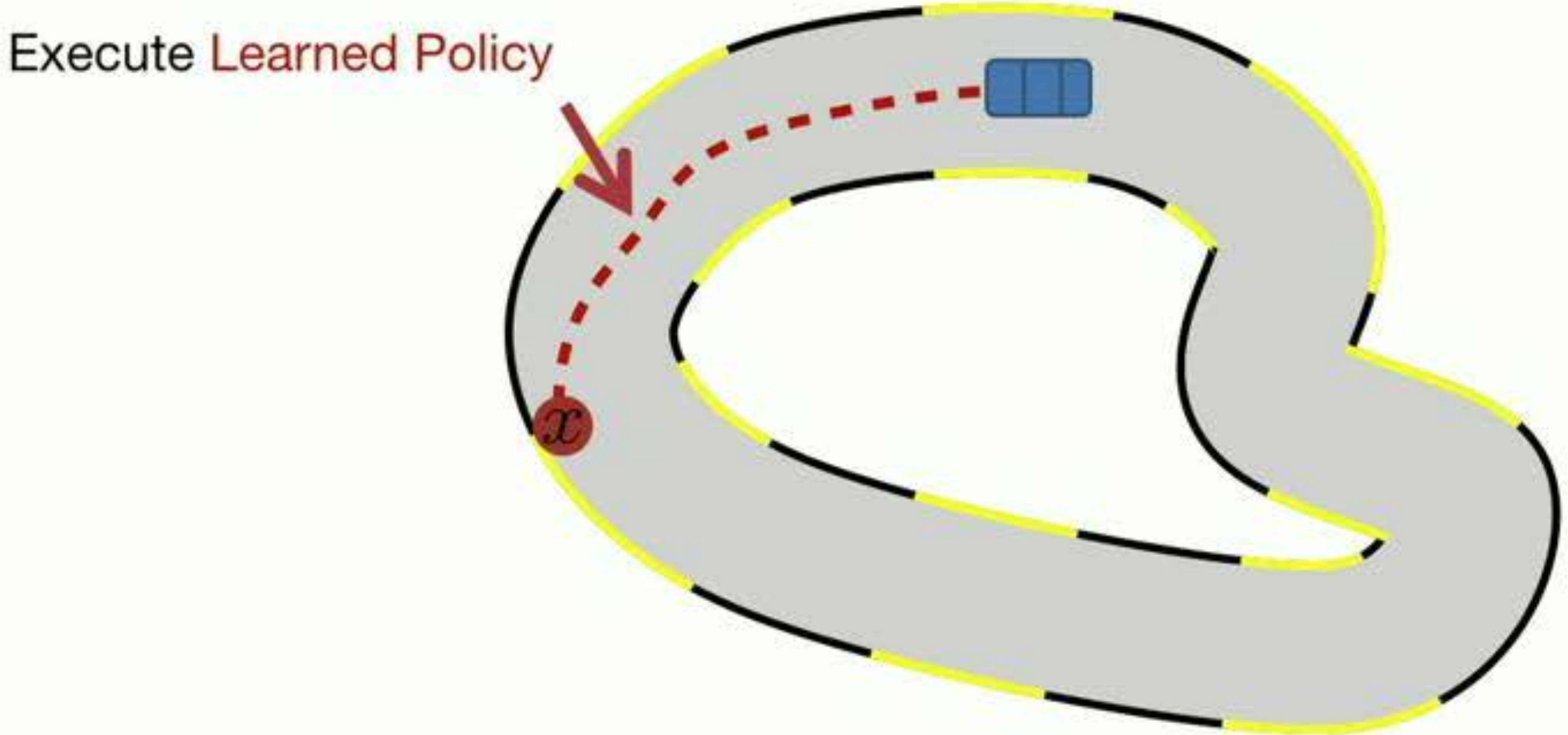
Interactive Imitation Learning w/ Reward

A global expert is available during training



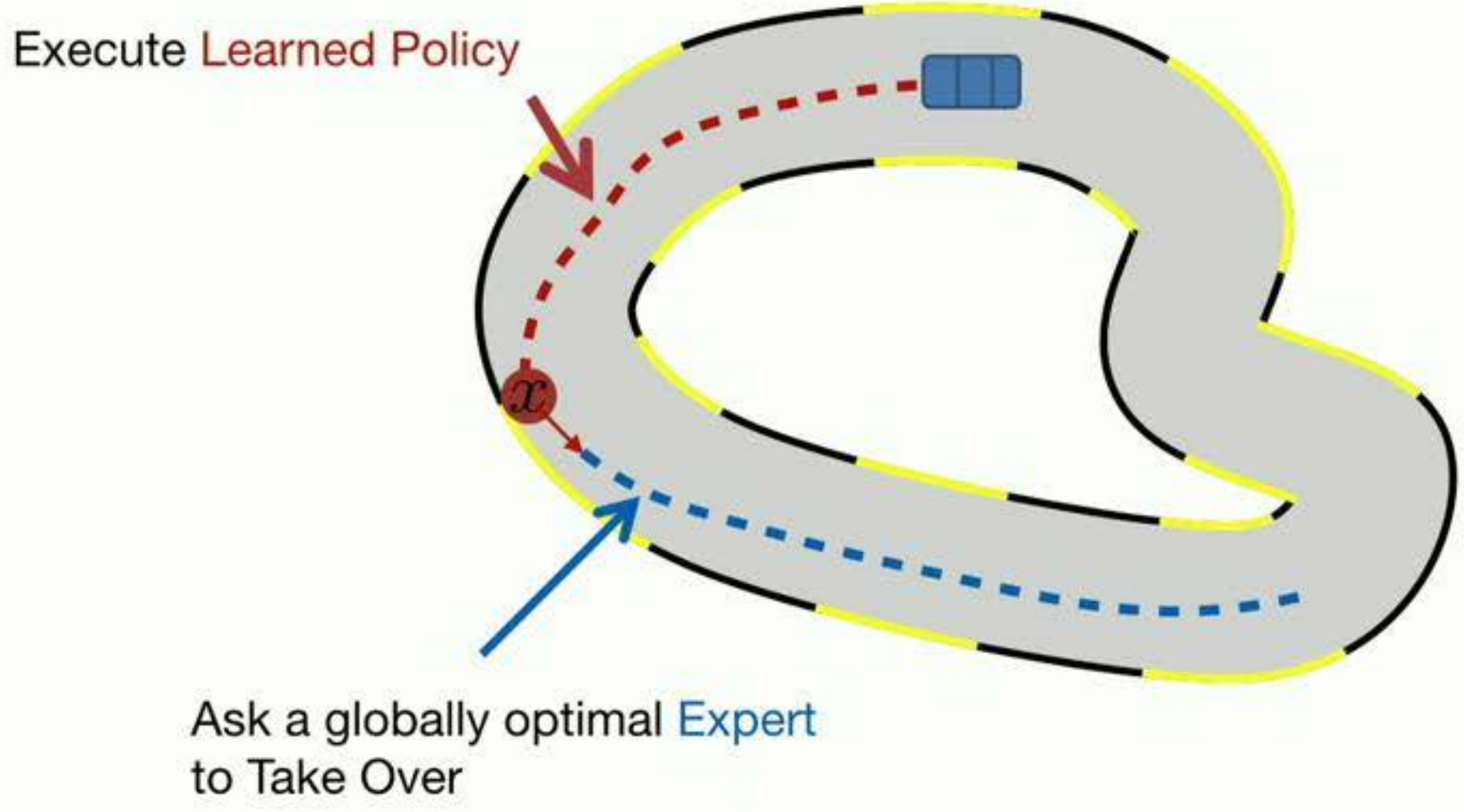
Interactive Imitation Learning w/ Reward

A global expert is available during training



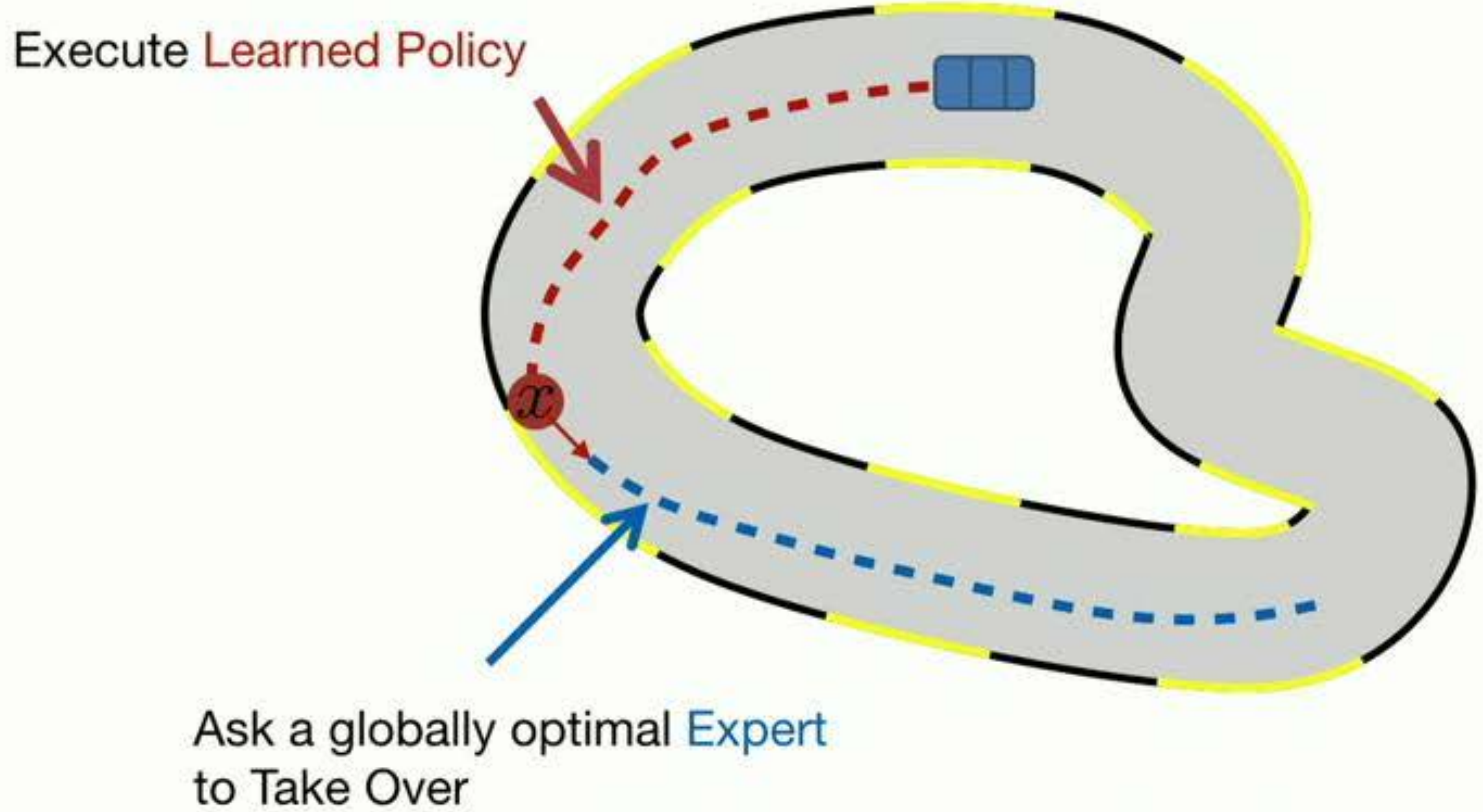
Interactive Imitation Learning w/ Reward

A global expert is available during training



Interactive Imitation Learning w/ Reward

A global expert is available during training



Record: Expert trajectory's total cost

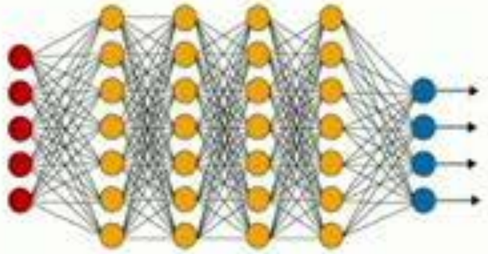
How easy to recover from the learner's mistake

Examples of Interactive Experts

Examples of Interactive Experts

1. Planner/Control (e.g., Robotics)

Cheap sensors

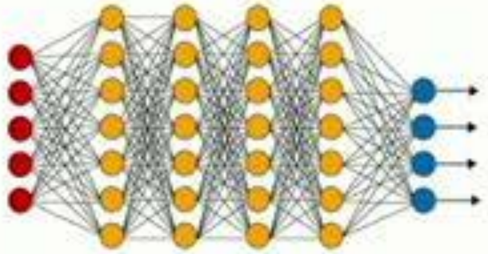


Controls signals

Examples of Interactive Experts

1. Planner/Control (e.g., Robotics)

Cheap sensors



Controls signals

Control w/
Expensive Sensors
(e.g., IMU & GPS)

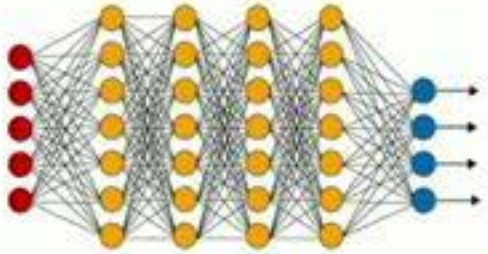
A global expert that
exists only at training

[Pan et.al, 17, RSS]

Examples of Interactive Experts

1. Planner/Control (e.g., Robotics)

Cheap sensors



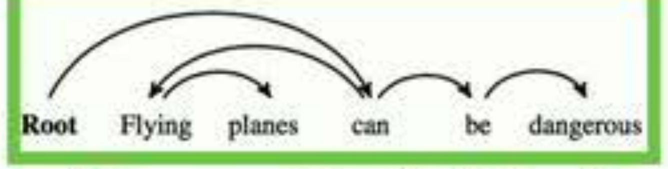
Controls signals

Control w/
Expensive Sensors
(e.g., IMU & GPS)

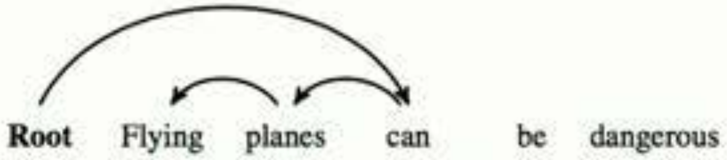
A global expert that exists only at training

[Pan et.al, 17, RSS]

2. Search Algorithms (e.g., NLP)

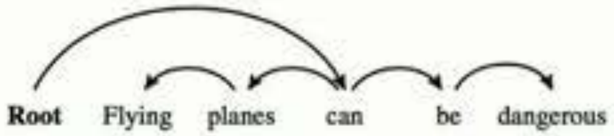


(Ground truth label)

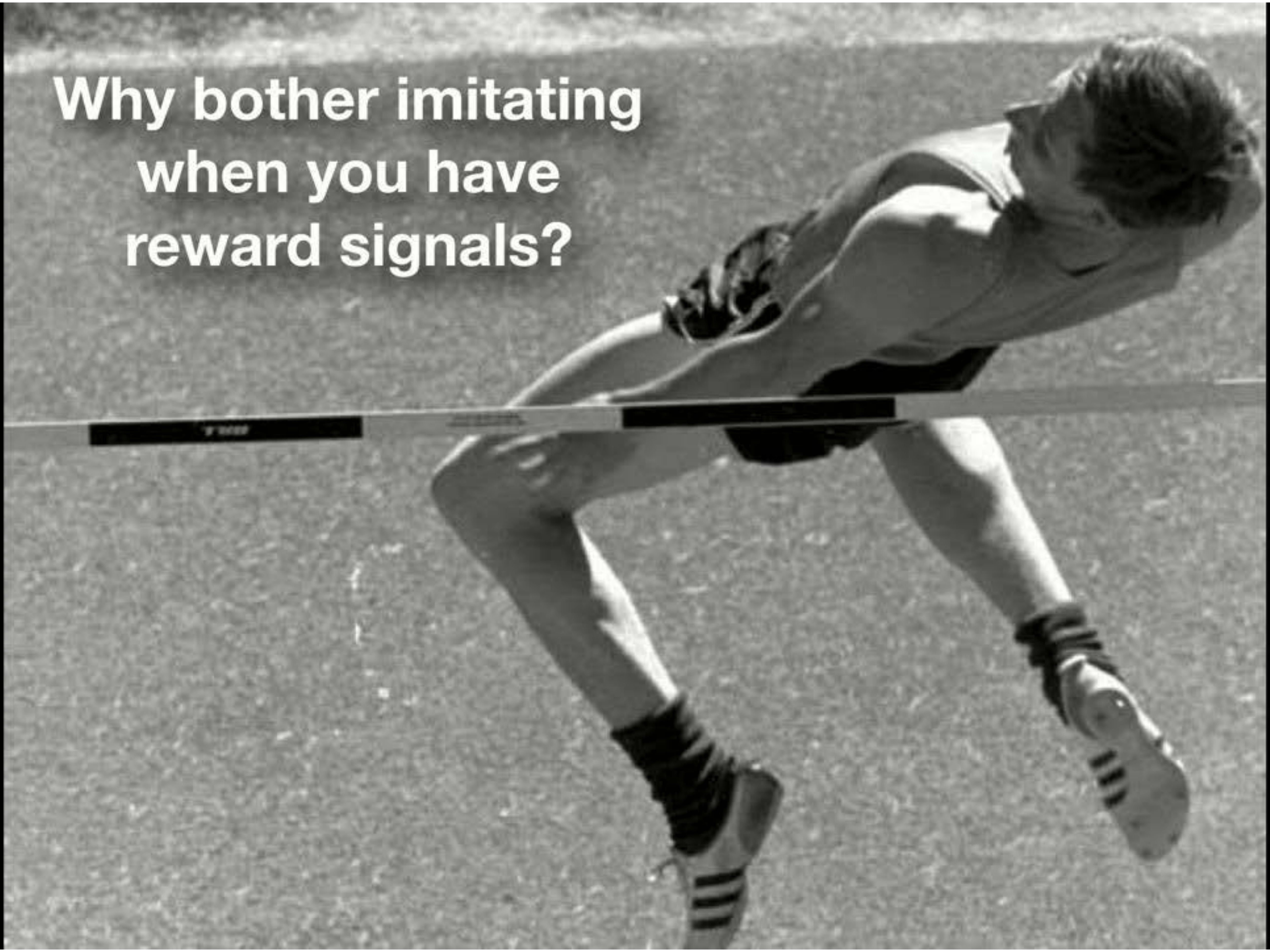


where I am…

Search to minimize Utility (e.g., F1)



[Goldberg & Nivre 12]
[Chang et.al., 15, ICML]


Why bother imitating
when you have
reward signals?

Why IL: Formalizing Advantages

1. Global Optimality

Global Optimal Expert: $\pi^\star$

AggreVaTe (Aggregate with Values) [Ross&Bagnell14]

$$J(\hat{\pi}) \approx J(\pi^\star)$$

Why IL: Formalizing Advantages

1. Global Optimality

Global Optimal Expert: $\pi^\star$

AggreVaTe (Aggregate with Values) [Ross&Bagnell14]

$$J(\hat{\pi}) \approx J(\pi^\star)$$

2. Sample Efficiency (i.e., Learns faster)

There exist MDPs, s.t. with global optimal expert, to learn near-optimal solution,

IL (e.g., AggreVaTe) $O(\log(S))$ **vs** **ANY** RL $\Omega(S)$

Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction
Sun, Venkatraman, Gordon, Boots, Bagnell, ICML, 17

Why IL: Formalizing Advantages

1. Global Optimality

Global Optimal Expert: $\pi^\star$

AggreVaTe (Aggregate with Values) [Ross&Bagnell14]

$$J(\hat{\pi}) \approx J(\pi^\star)$$

2. Sample Efficiency (i.e., Learns faster)

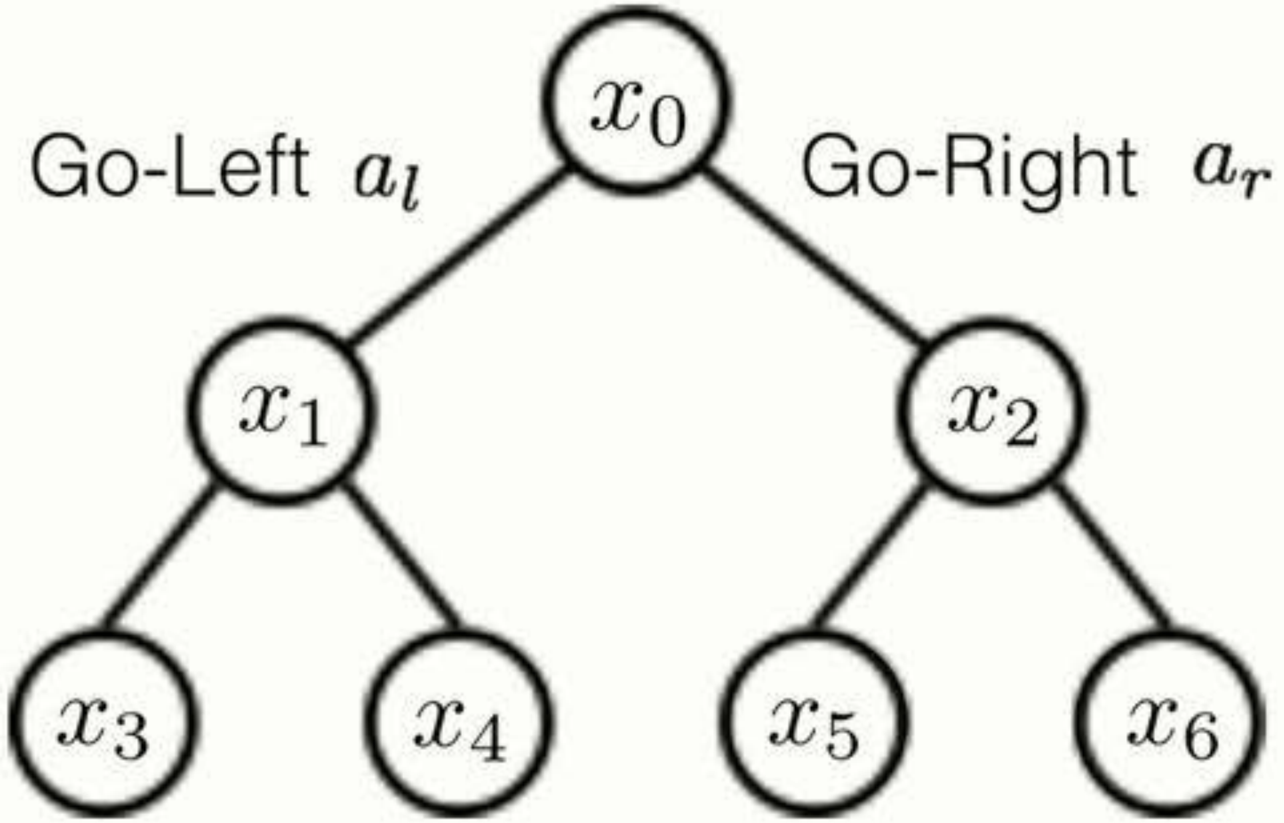
There exist MDPs, s.t. with global optimal expert, to learn near-optimal solution,

✓ **IL** (e.g., AggreVaTe) $O(\log(S))$ **vs** **ANY** RL $\Omega(S)$

Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction
Sun, Venkatraman, Gordon, Boots, Bagnell, ICML, 17

Deterministic MDP

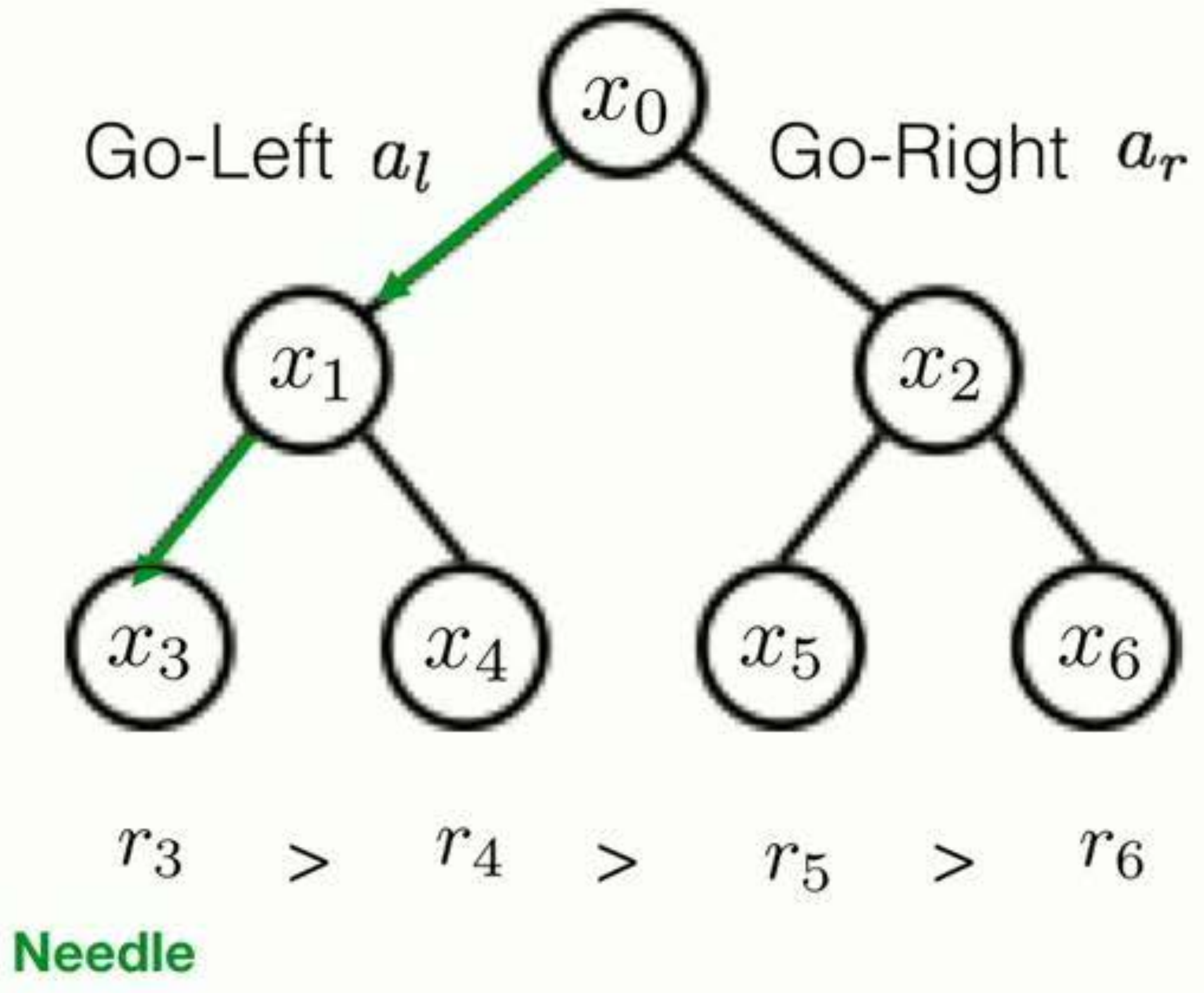
Global Optimal Expert: An Optimal Planner



$r_3 > r_4 > r_5 > r_6$

Deterministic MDP

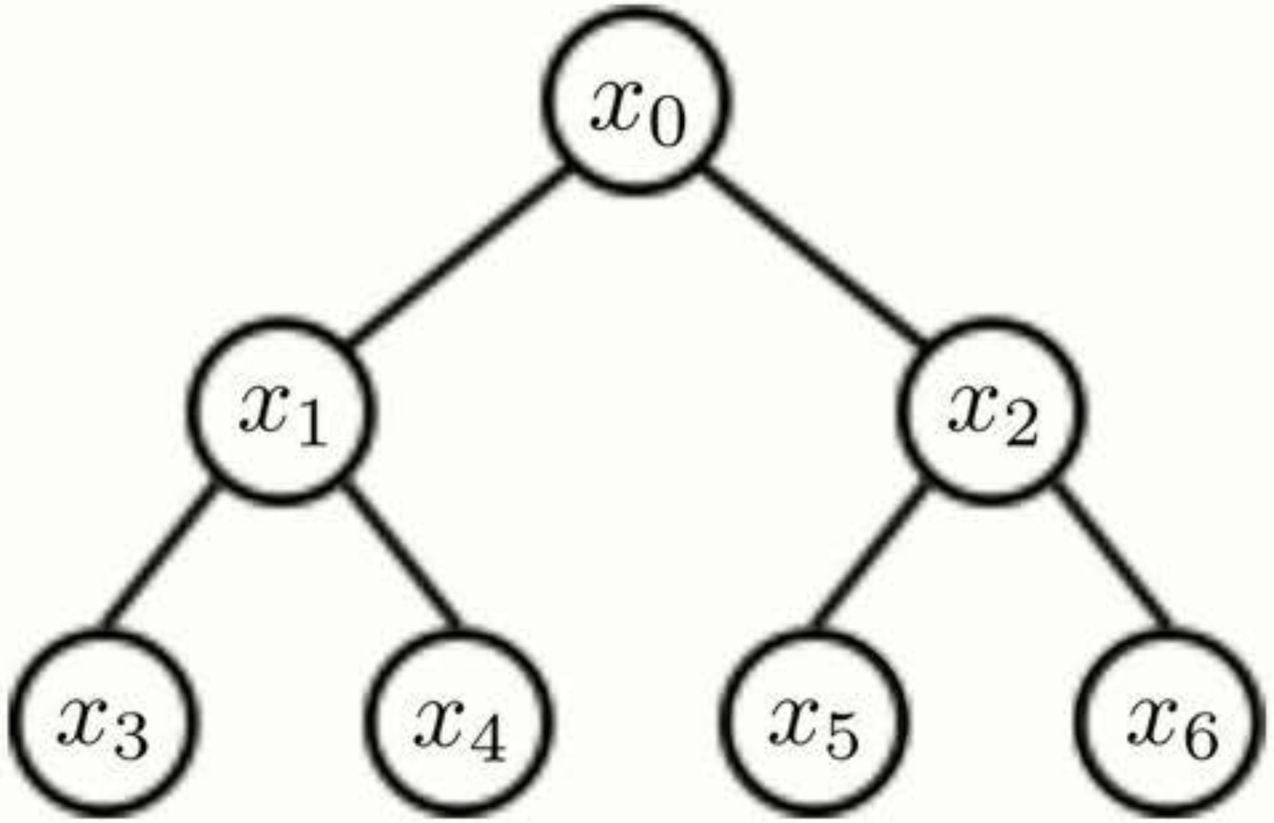
Global Optimal Expert: An Optimal Planner



Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$$r_3 > r_4 > r_5 > r_6$$

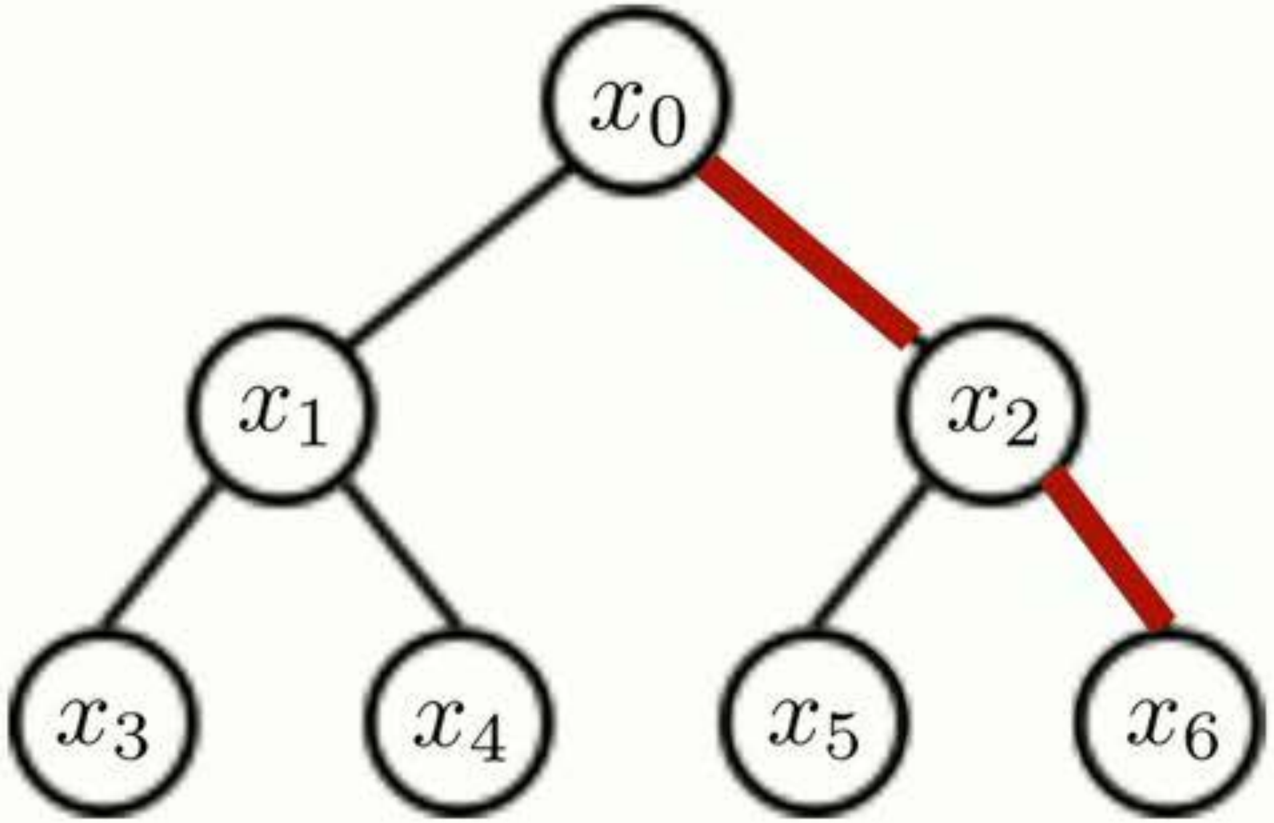
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$r_3 > r_4 > r_5 > r_6$

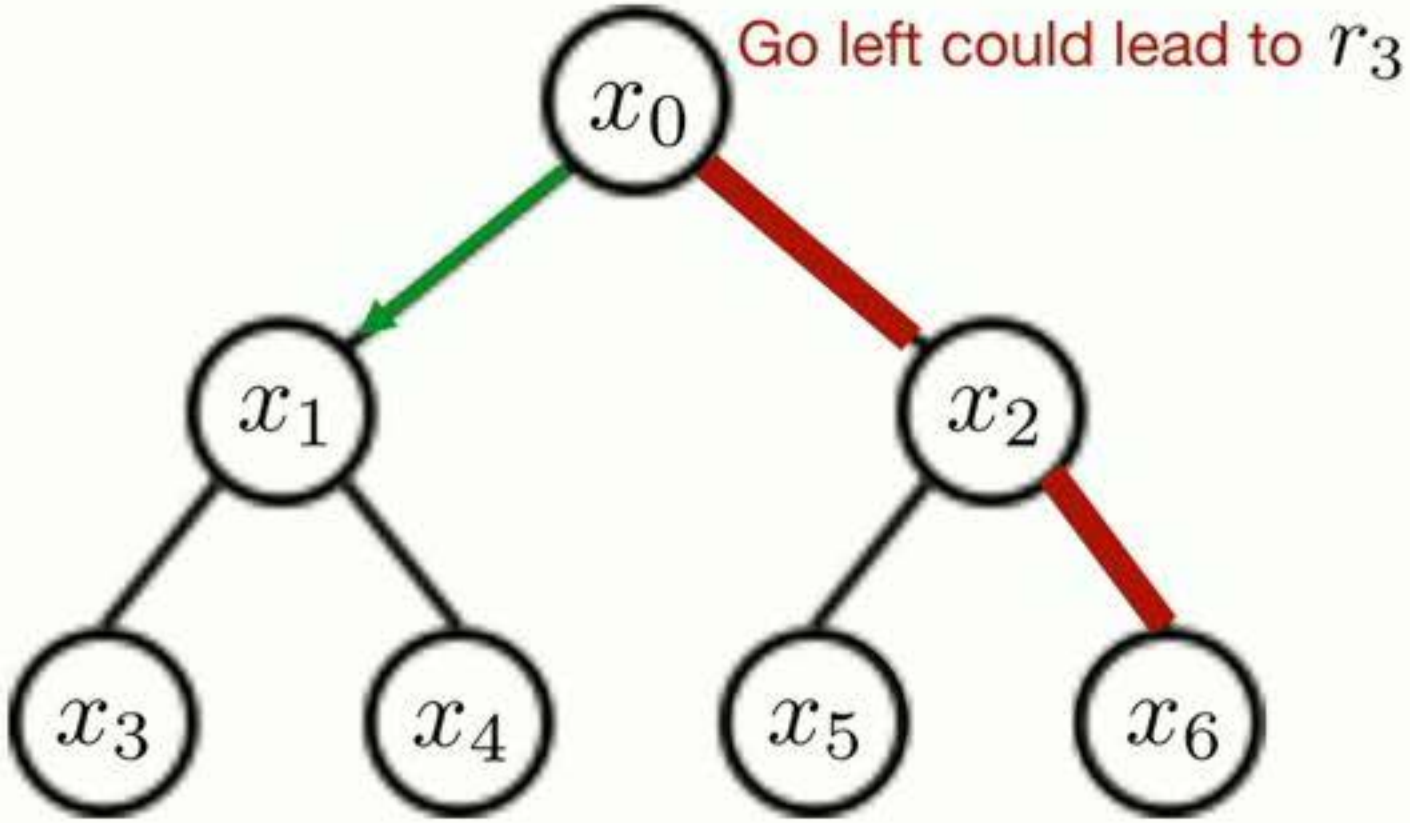
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$r_3 > r_4 > r_5 > r_6$

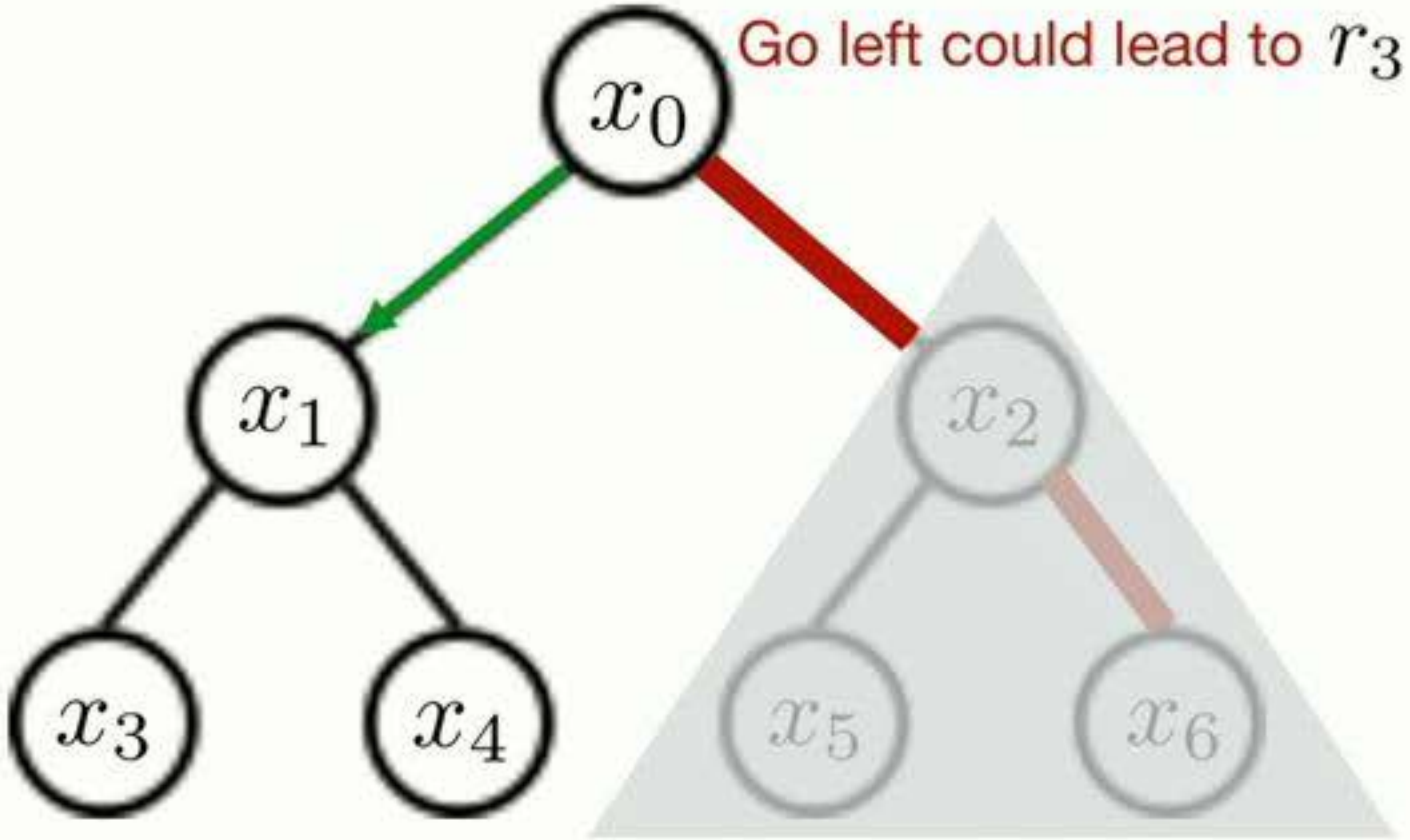
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$r_3 > r_4 > r_5 > r_6$

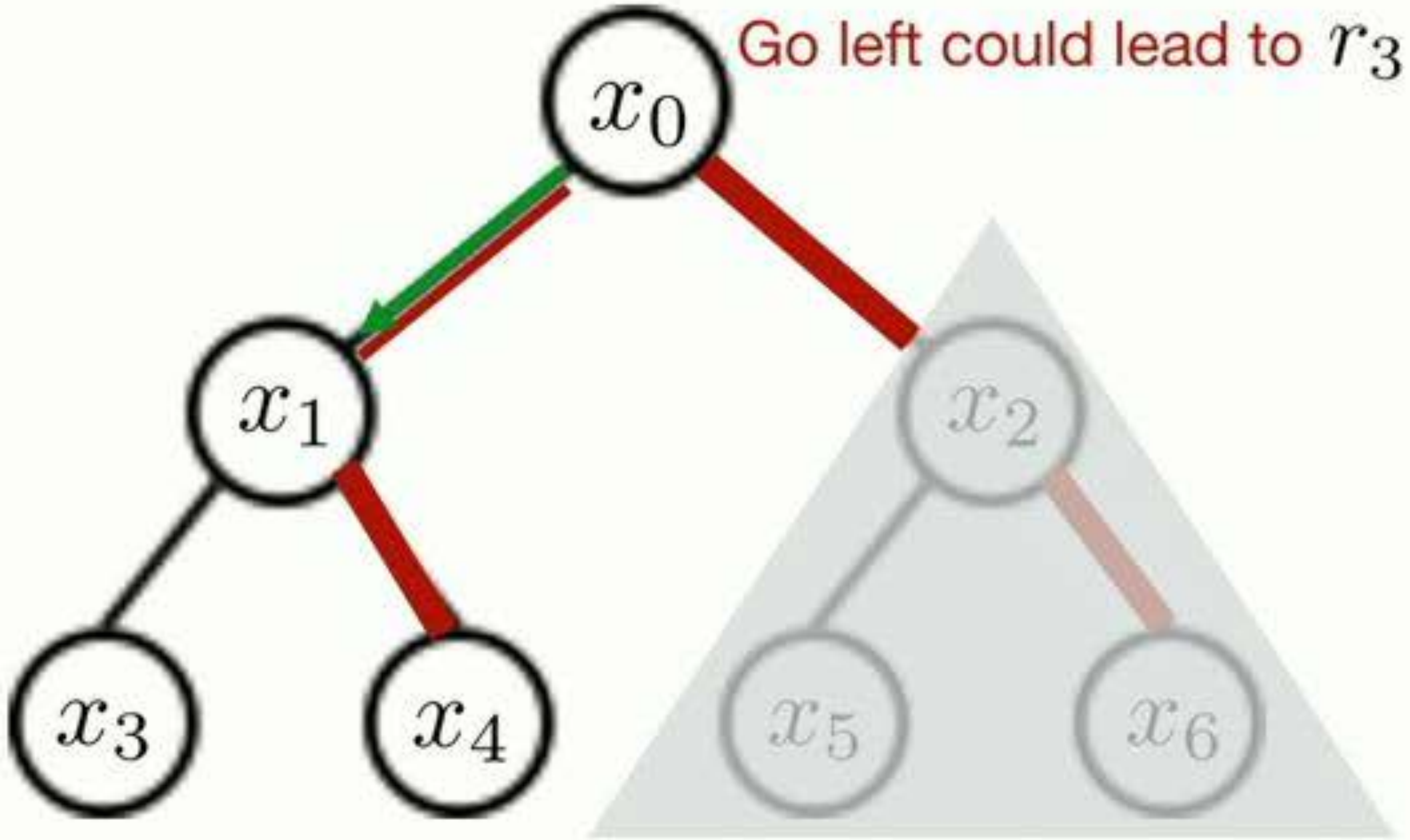
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$r_3 > r_4 > r_5 > r_6$

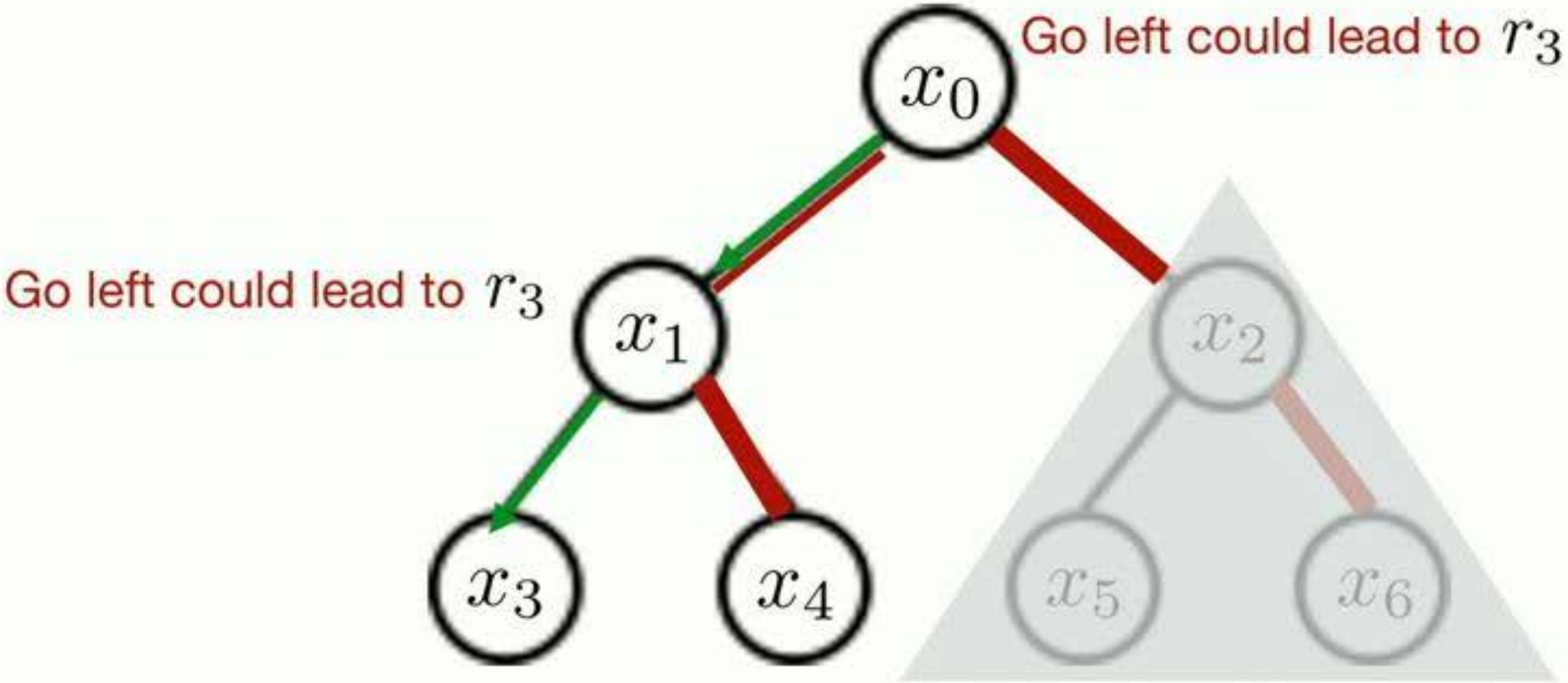
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$$r_3 > r_4 > r_5 > r_6$$

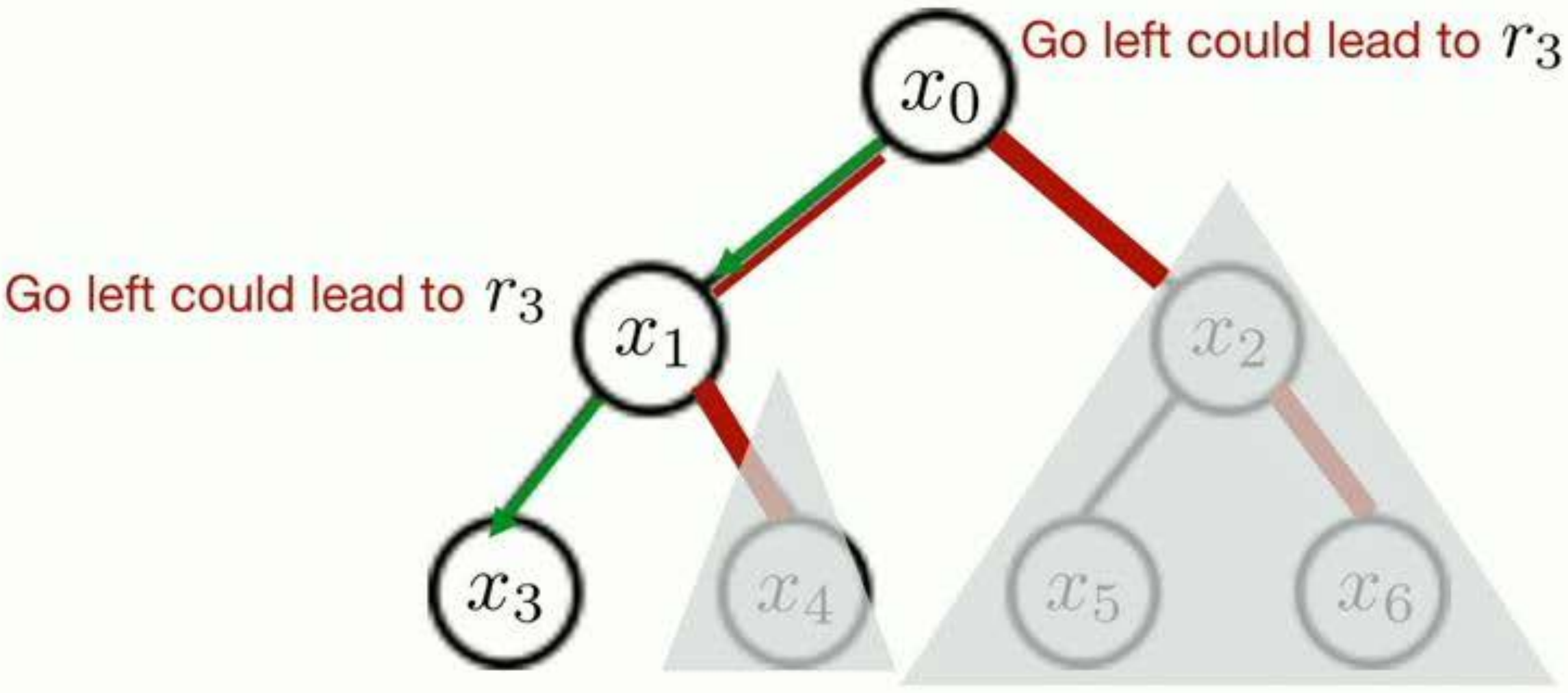
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



$$r_3 > r_4 > r_5 > r_6$$

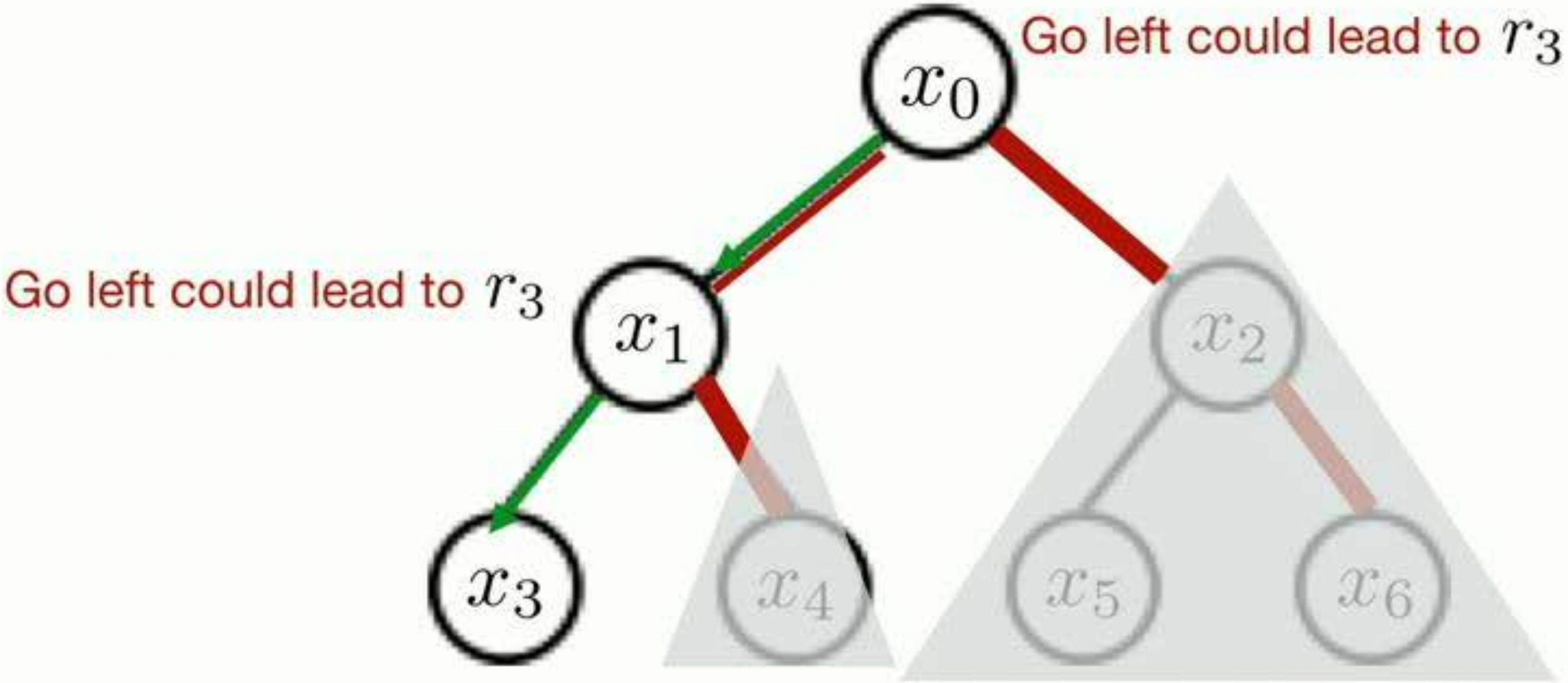
Halving: Eliminate half of the nodes each round

[**Sun**,et.al., 17, ICML]

Reduction to Supervised Learning

Easy Credit Assignment

Global Optimal Expert: An Optimal Planner



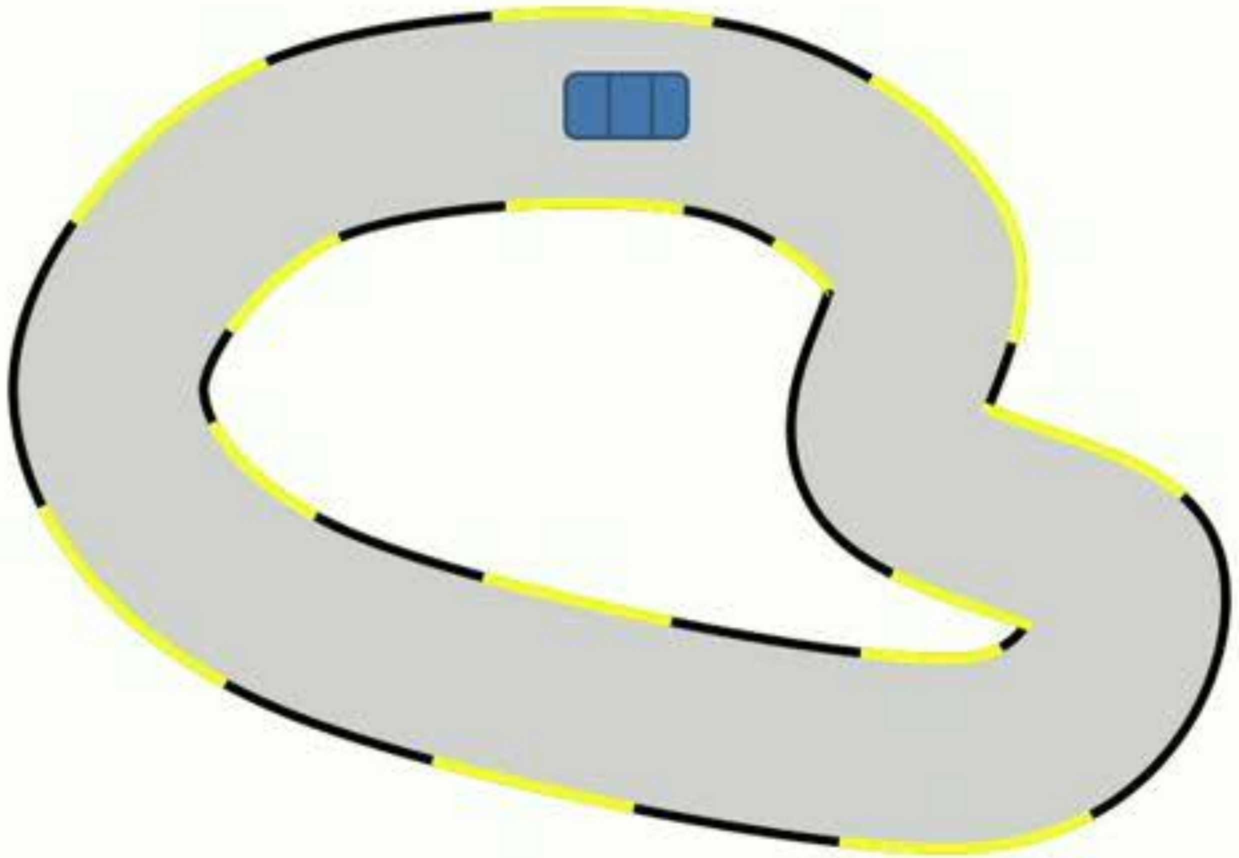
$$r_3 > r_4 > r_5 > r_6$$

Halving: Eliminate half of the nodes each round

IL: $\log(S)$ **vs** **RL:** $\Omega(S)$

[**Sun**,et.al., 17, ICML]

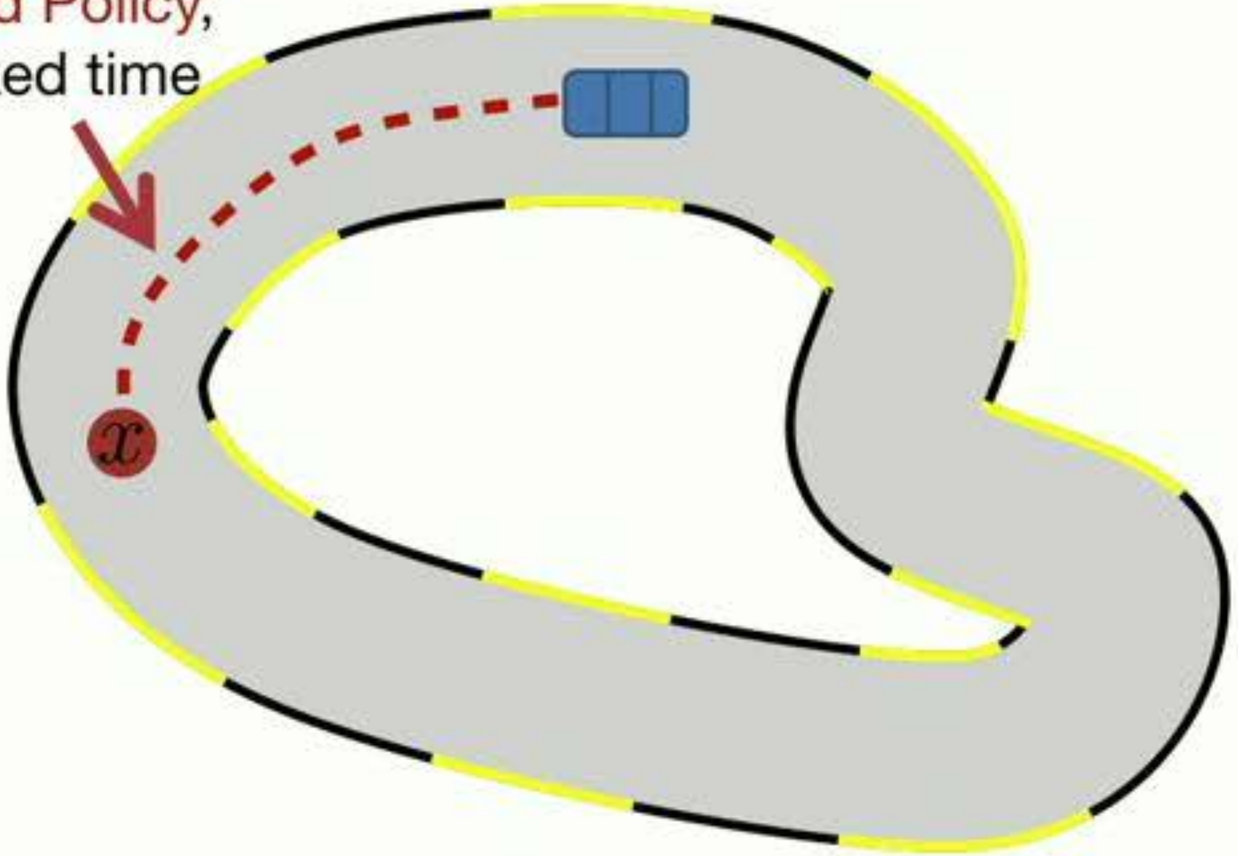
Ex: AggreVaTe [Ross & Bagnell, 14]



Ex: AggreVaTe

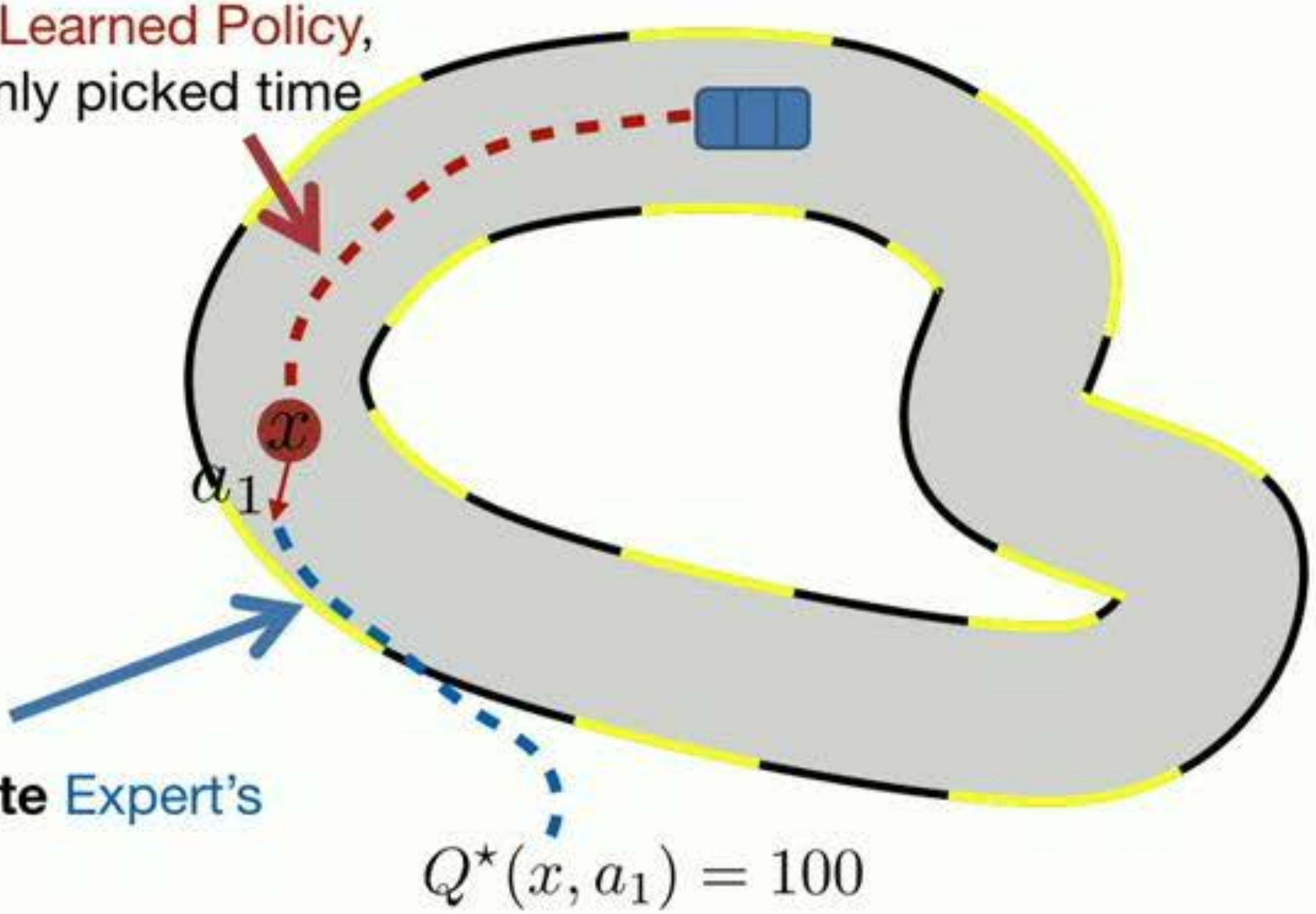
[Ross & Bagnell, 14]

Rollin: Execute Learned Policy,
Stop at a randomly picked time step



Ex: AggreVaTe [Ross & Bagnell, 14]

Rollin: Execute Learned Policy,
Stop at a randomly picked time step

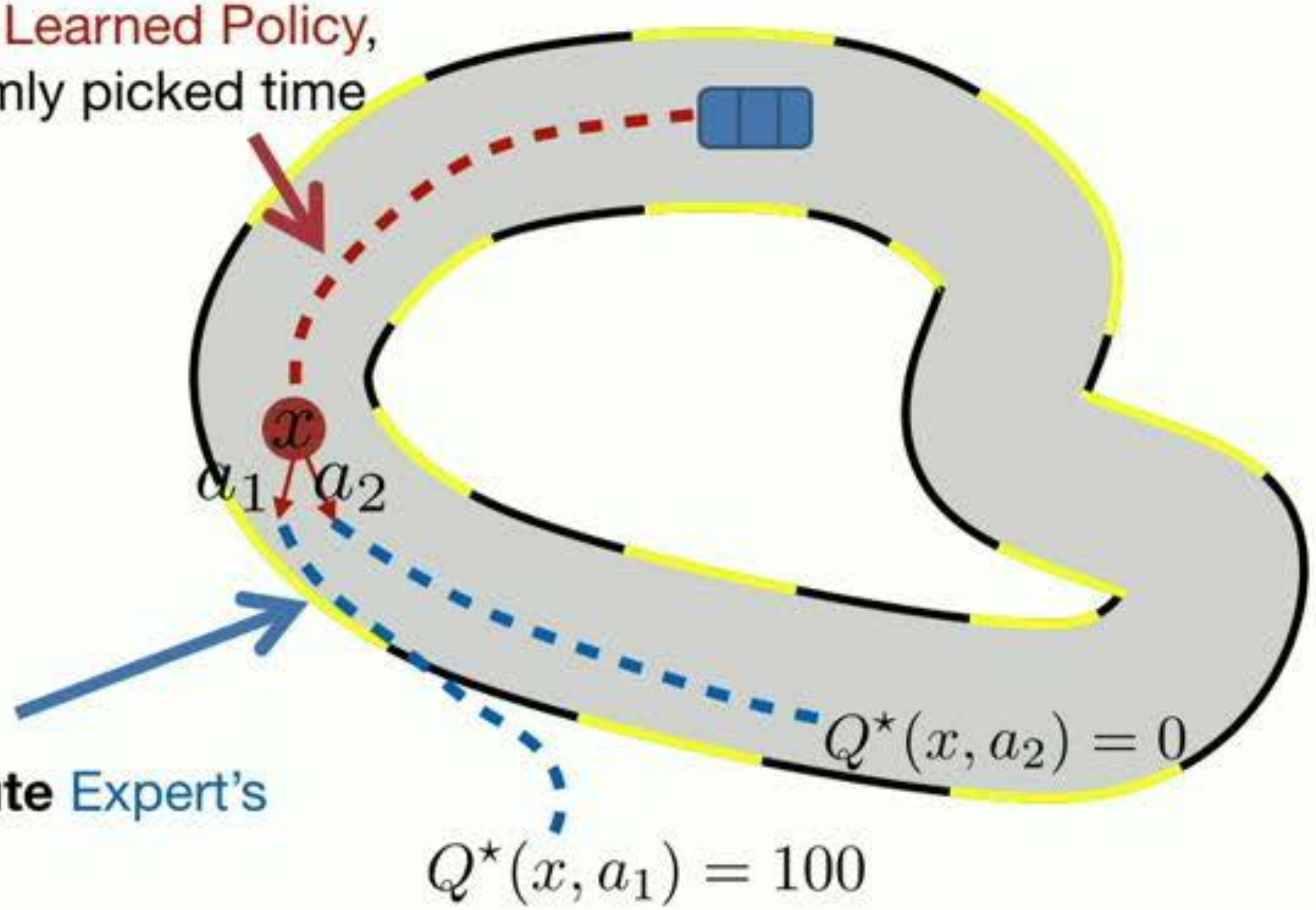


Rollout: Execute Expert's Policy

Ex: AggreVaTe

[Ross & Bagnell, 14]

Rollin: Execute Learned Policy,
Stop at a randomly picked time step

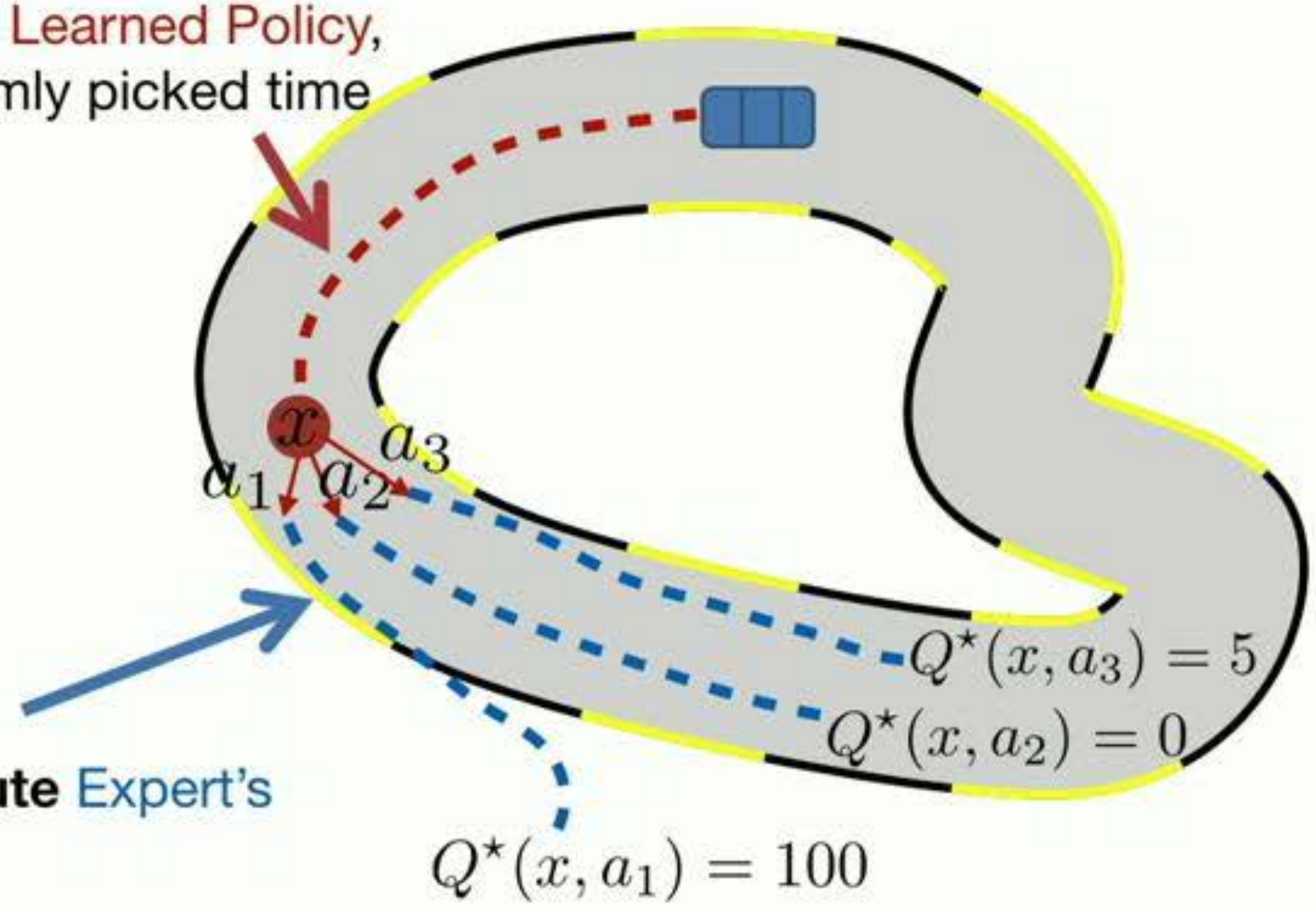


Rollout: Execute Expert's Policy

Ex: AggreVaTe

[Ross & Bagnell, 14]

Rollin: Execute Learned Policy, Stop at a randomly picked time step

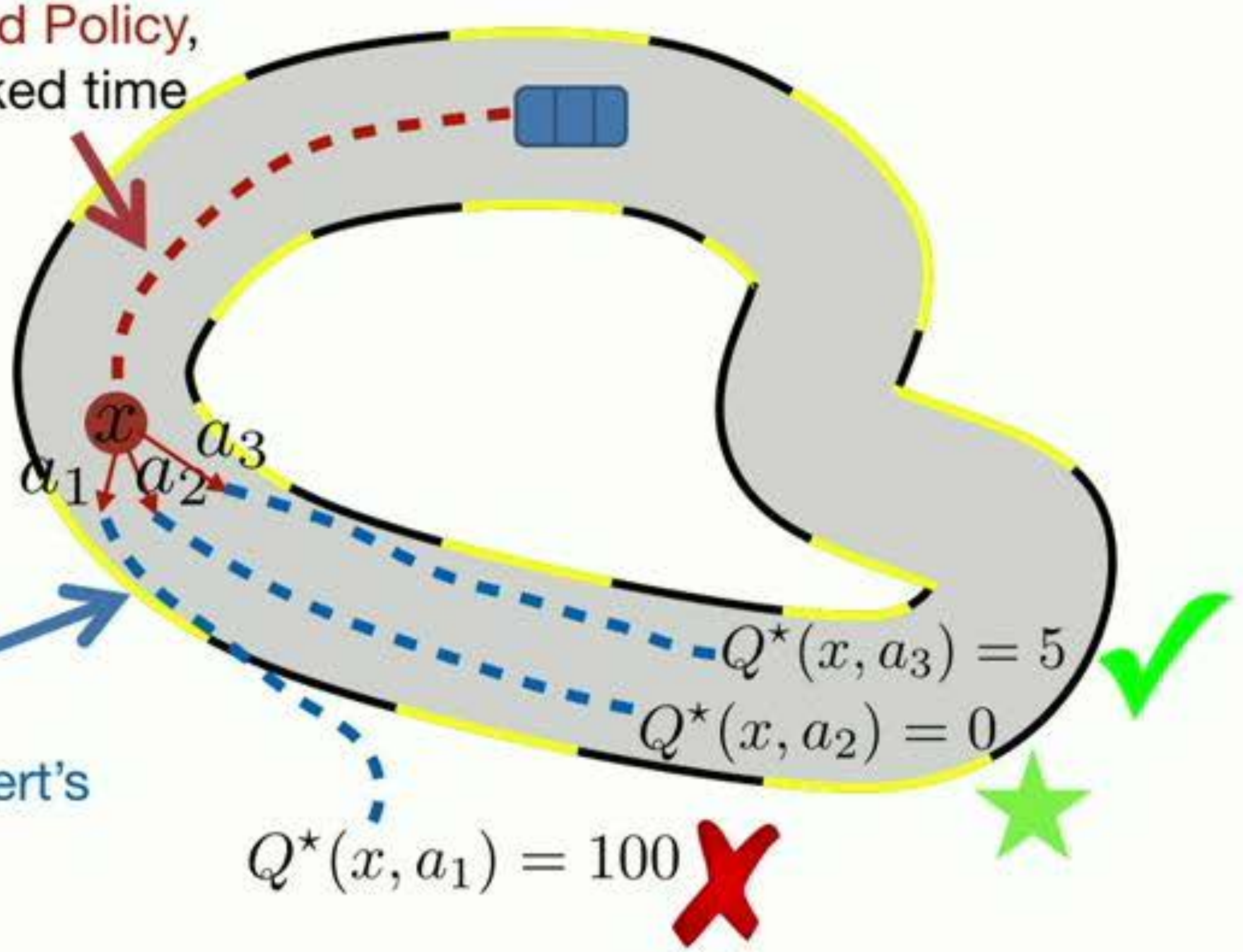


Rollout: Execute Expert's Policy

Ex: AggreVaTe

[Ross & Bagnell, 14]

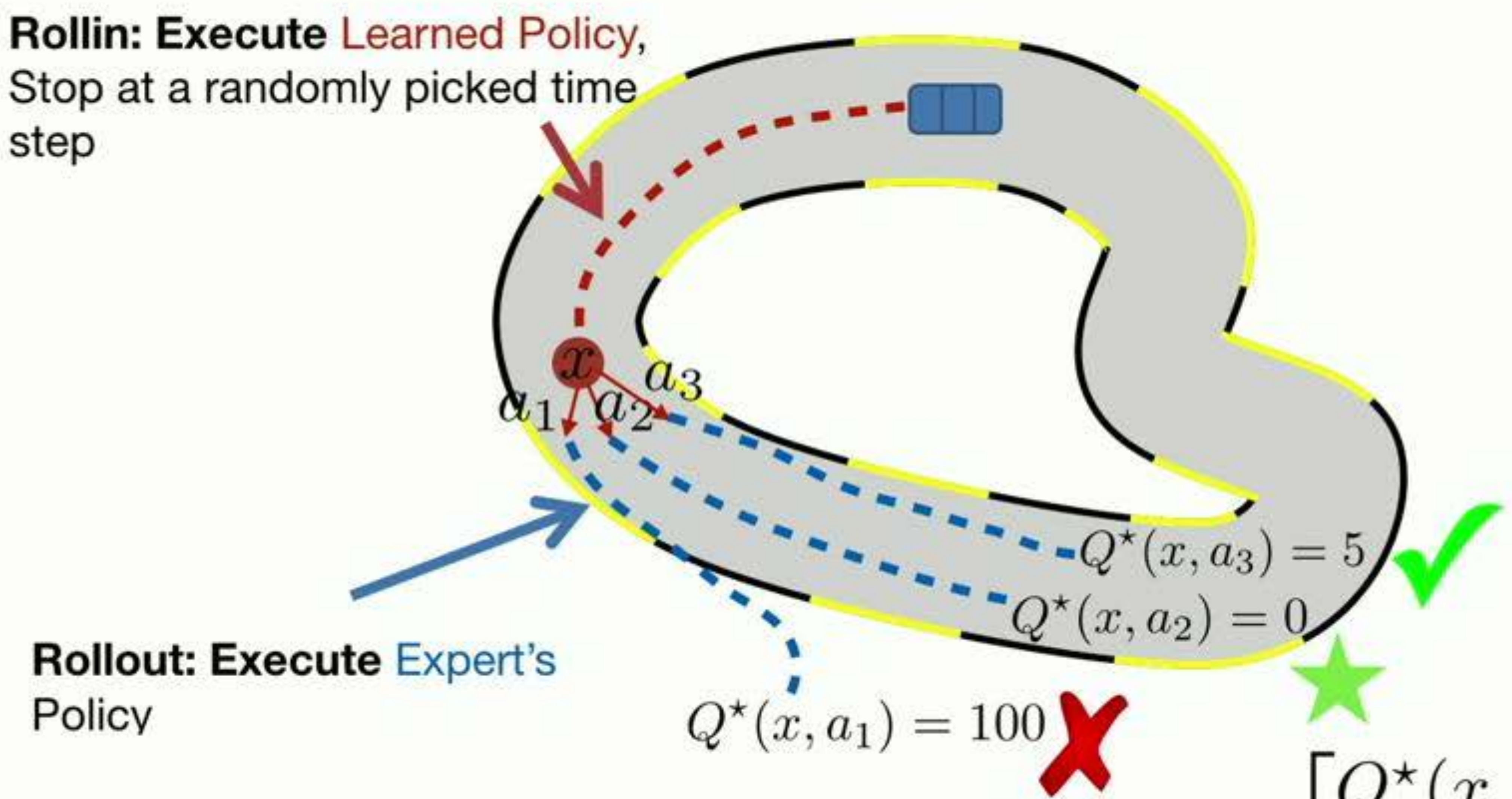
Rollin: Execute Learned Policy,
Stop at a randomly picked time step



Rollout: Execute Expert's
Policy

Ex: AggreVaTe

[Ross & Bagnell, 14]



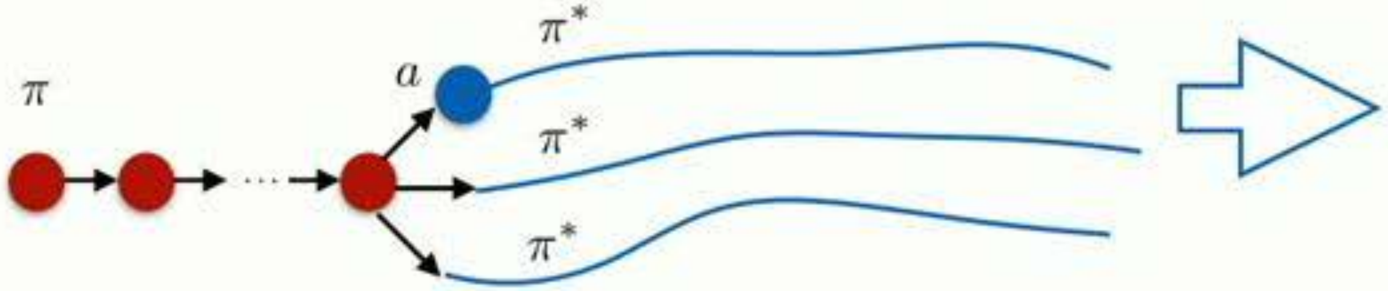
$$\left\{ x, \begin{bmatrix} Q^\star(x, a_1) \\ Q^\star(x, a_2) \\ Q^\star(x, a_3) \end{bmatrix} \right\}_N$$

Cost-Sensitive Classification Dataset
(A **Supervised Learning** Dataset)

Differentiable AggreVaTe (AggreVaTe**D**)

[**Sun**,et.al., 17, ICML]

π_{θ_n} (e.g., neural net)



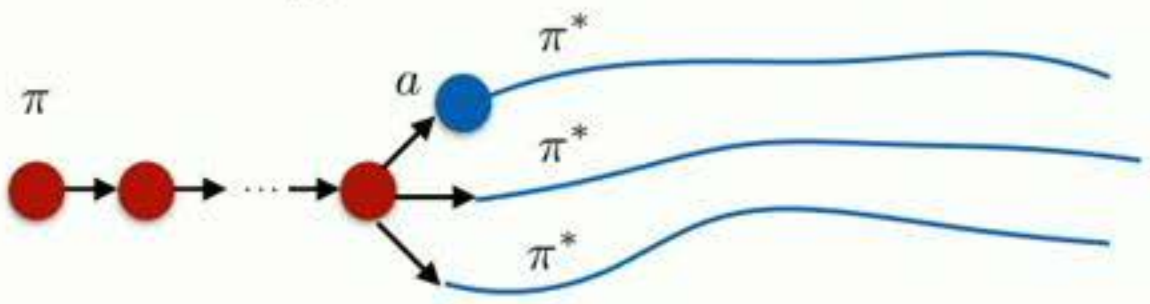
$$\left\{x, \begin{bmatrix} Q^\star(x, a_1) \\ Q^\star(x, a_2) \\ Q^\star(x, a_3) \end{bmatrix} \right\}_N$$

Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction
[**Sun**, Venketraman, Gordon, Boots, Bagnell, ICML, 17]

Differentiable AggreVaTe (AggreVaTe**D**)

[**Sun**,et.al., 17, ICML]

π_{θ_n} (e.g., neural net)



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

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

$$\ell_n(\theta) = \sum_i \sum_a \pi(a|x_i;\theta) Q^\star(x_i, a)$$

Cost-Sensitive loss

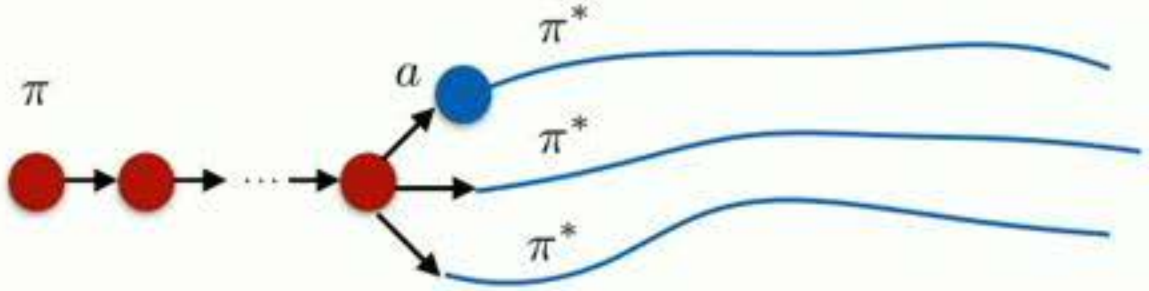
Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction
[**Sun**, Venketraman, Gordon, Boots, Bagnell, ICML, 17]

Differentiable AggreVaTe

[**Sun**,et.al., 17, ICML]

(AggreVaTe**D**)

π_{θ_n} (e.g., neural net)



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

$\pi_{\theta_{n+1}}$

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

e.g., Gradient or natural gradient descent

$$\ell_n(\theta) = \sum_i \sum_a \pi(a|x_i;\theta) Q^\star(x_i, a)$$

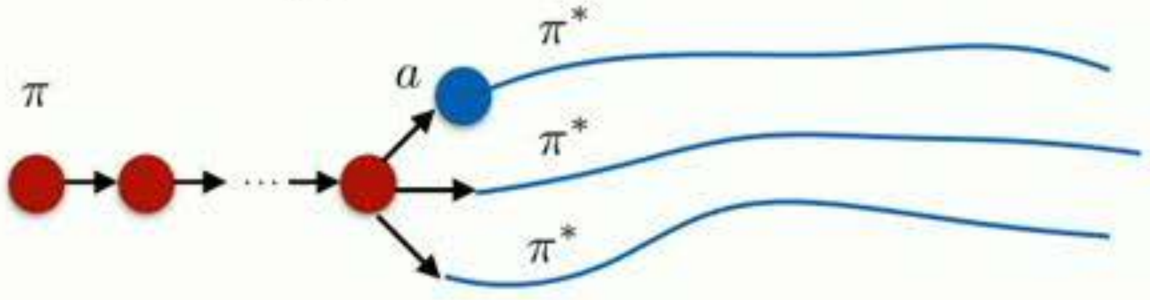
Cost-Sensitive loss

Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction
[**Sun**, Venketraman, Gordon, Boots, Bagnell, ICML, 17]

Differentiable AggreVaTe (AggreVaTe**D**)

[**Sun**,et.al., 17, ICML]

π_{θ_n} (e.g., neural net)



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

$$\ell_n(\theta) = \sum_i \sum_a \pi(a|x_i;\theta) Q^\star(x_i, a)$$

Cost-Sensitive loss

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

e.g., Gradient or natural gradient descent

$$\pi_{\theta_{n+1}}$$

Use rich function approximators for complex features

Deeply AggreVaTeD: Differential Imitation Learning for Sequential Prediction [**Sun**, Venketraman, Gordon, Boots, Bagnell, ICML, 17]

Dependency Parsing

Handwritten Algebra Equations & Solutions

[Duyck & Gordon 15]

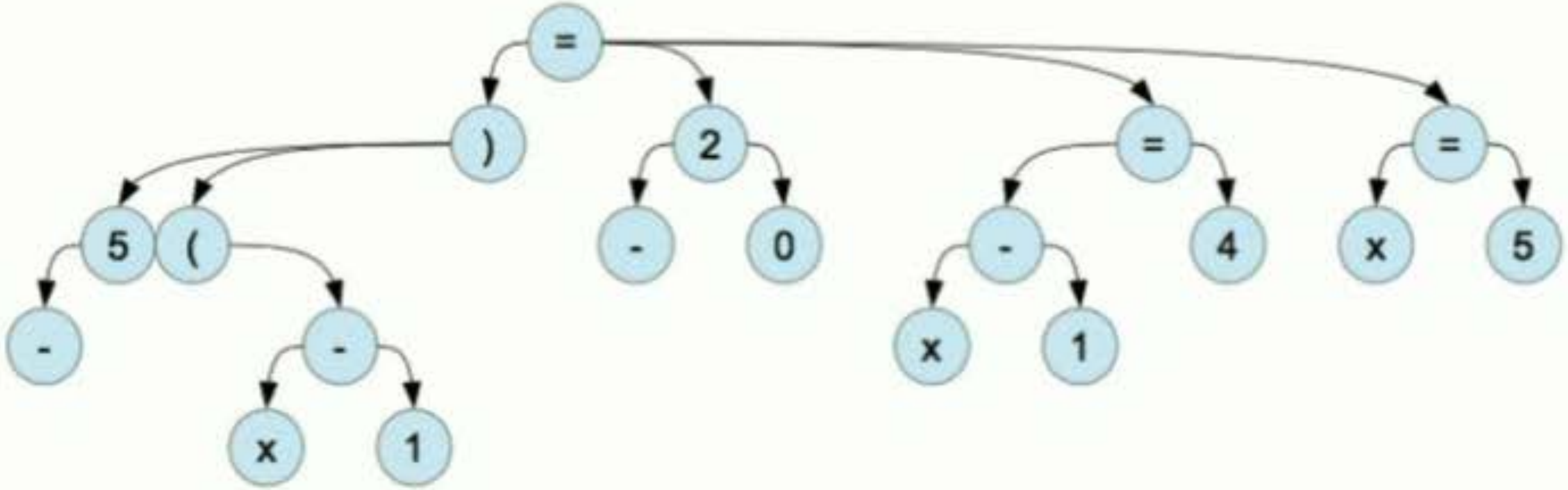
Input:

$$-5(x-1) = -20$$

$$x-1 = 4$$

$$x = 5$$

Output:



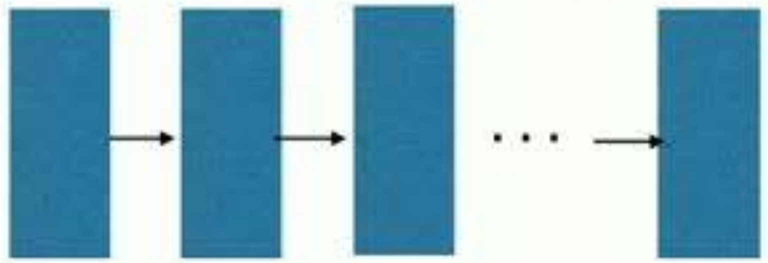
Dependency Parsing as Sequential Decision Making

[e.g., Chang, Krishnamurthy, Agarwal, Daume´ III, Langford, 15, ICML]

Dependency Parsing as Sequential Decision Making

[e.g., Chang, Krishnamurthy, Agarwal, Daume´ III, Langford, 15, ICML]

Encoder (LSTM)



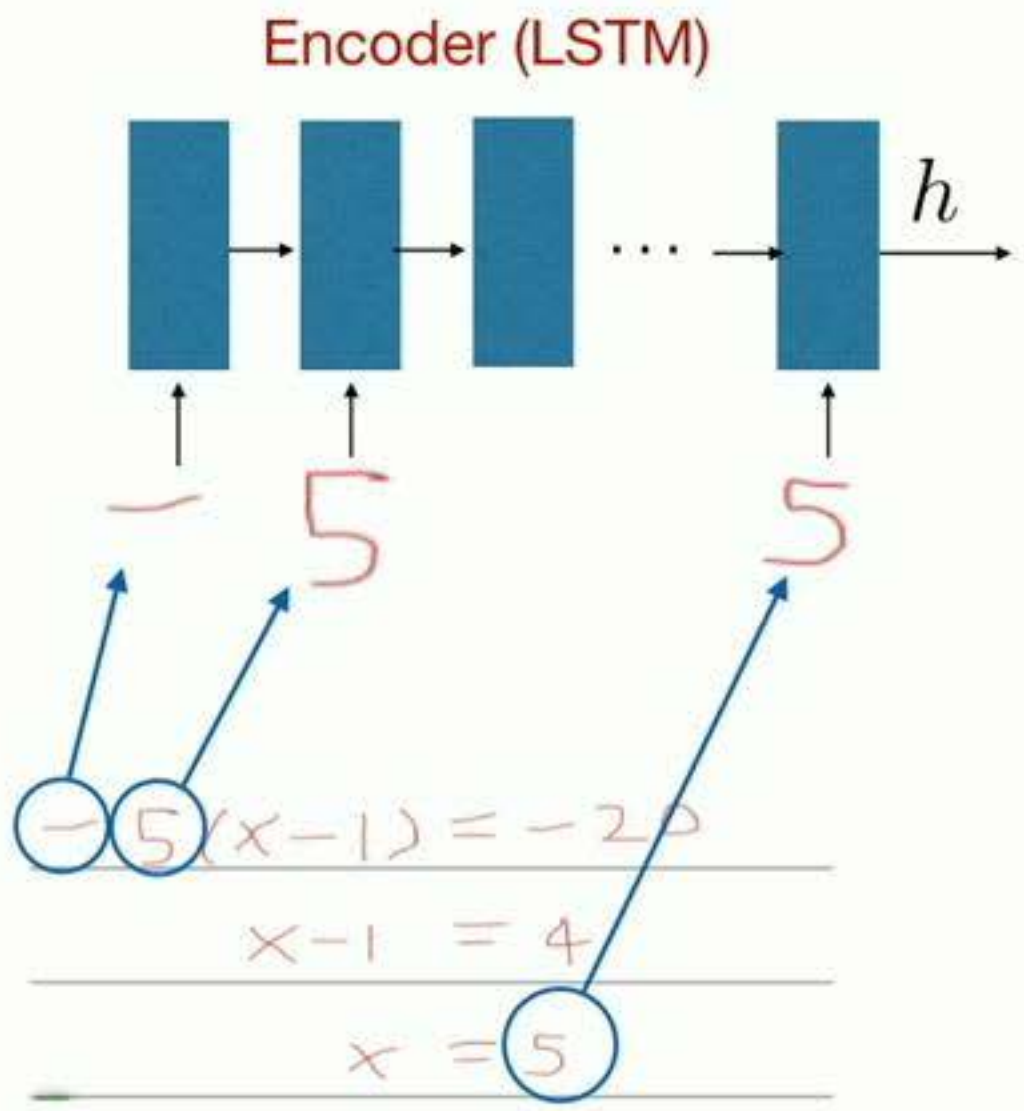
$$-5(x-1) = -20$$

$$x-1 = 4$$

$$x = 5$$

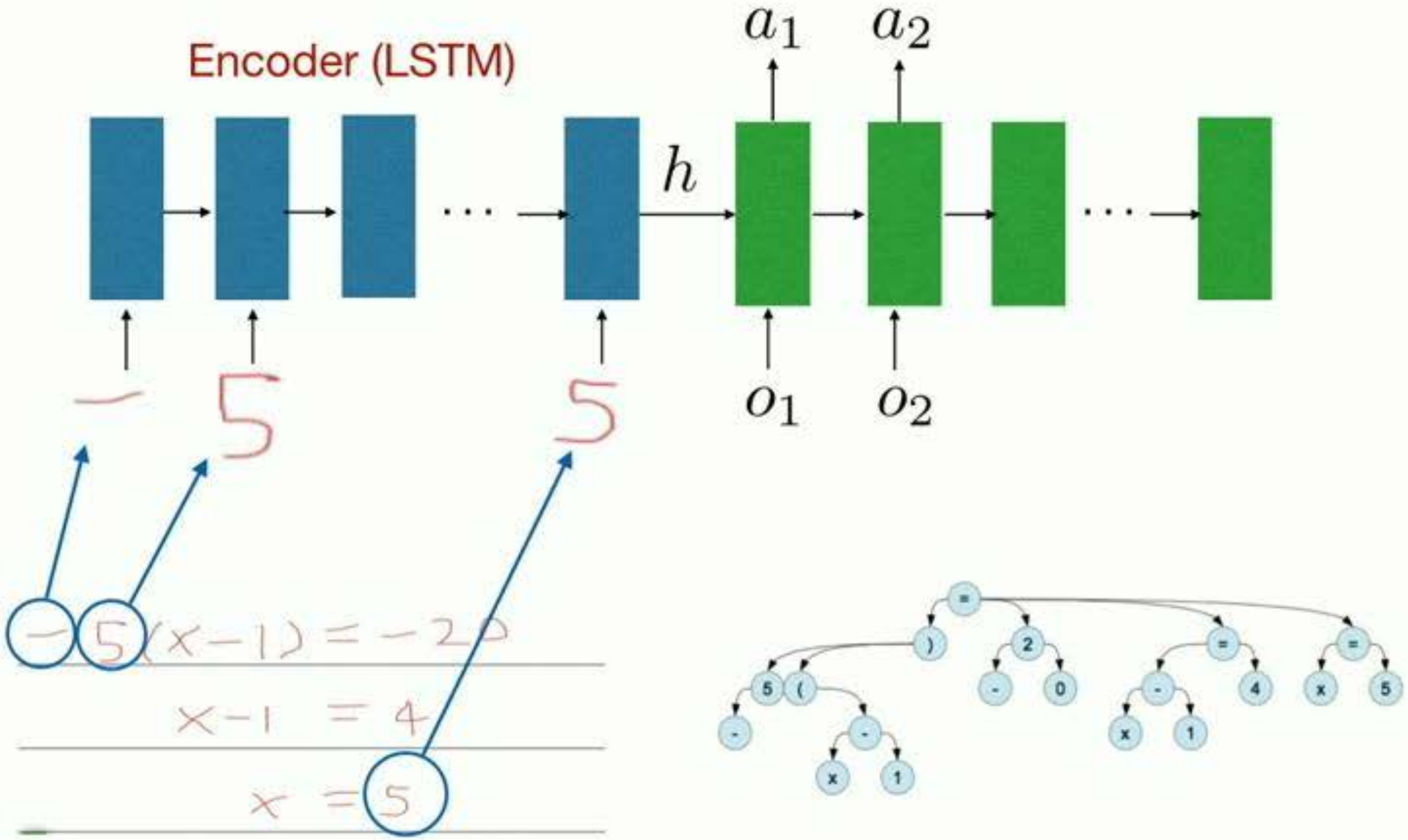
Dependency Parsing as Sequential Decision Making

[e.g., Chang, Krishnamurthy, Agarwal, Daume´ III, Langford, 15, ICML]

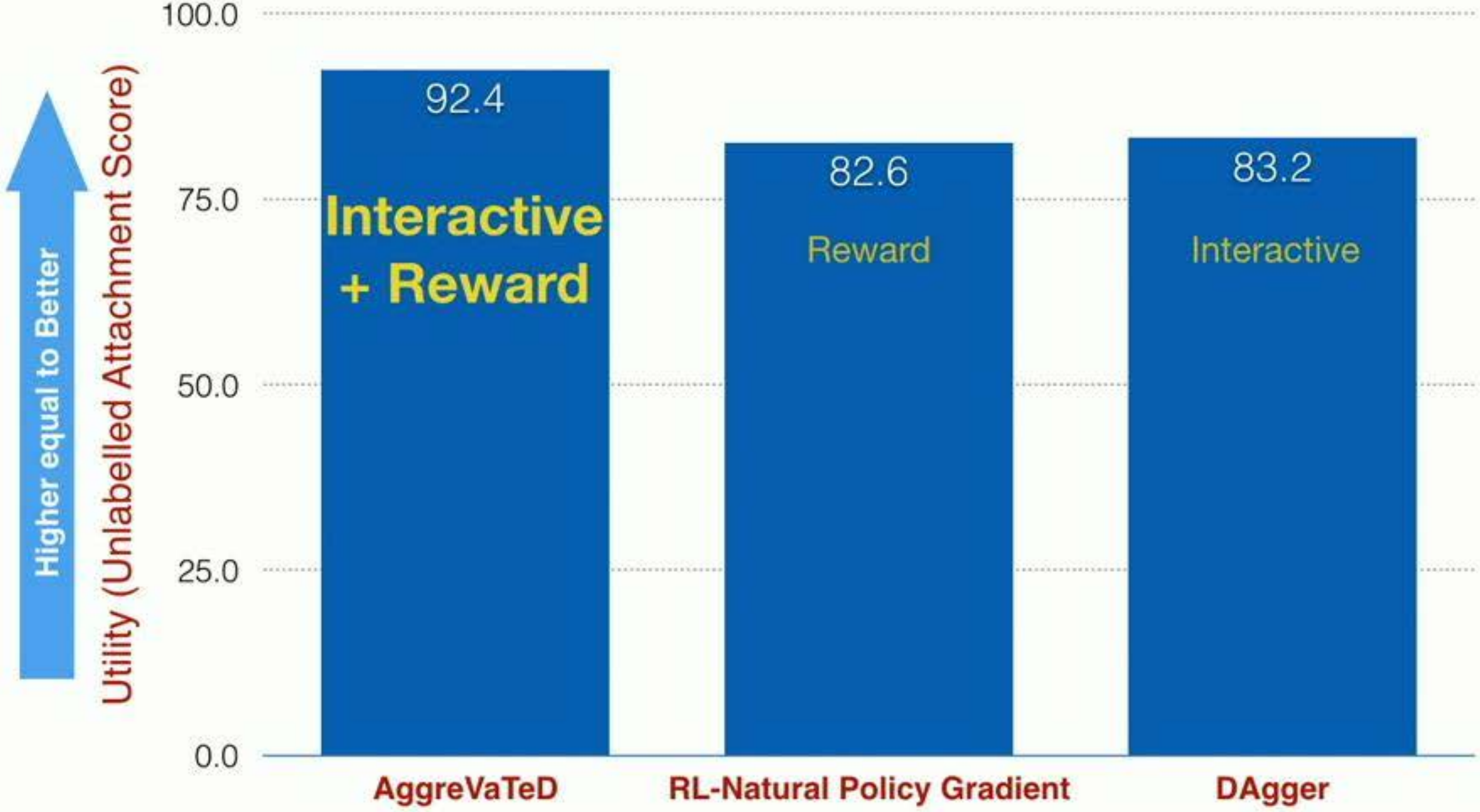


Dependency Parsing as Sequential Decision Making

[e.g., Chang, Krishnamurthy, Agarwal, Daume´ III, Langford, 15, ICML]



Performance of AggreVaTeD, RL, and DAgger



RL: Natural Policy Gradient [Kakade 02, NIPS, Bagnell, 04, IJCAI]
DAgger results from Duyck & Gordon, 15

What if we *do not* have a Globally Optimal Expert?

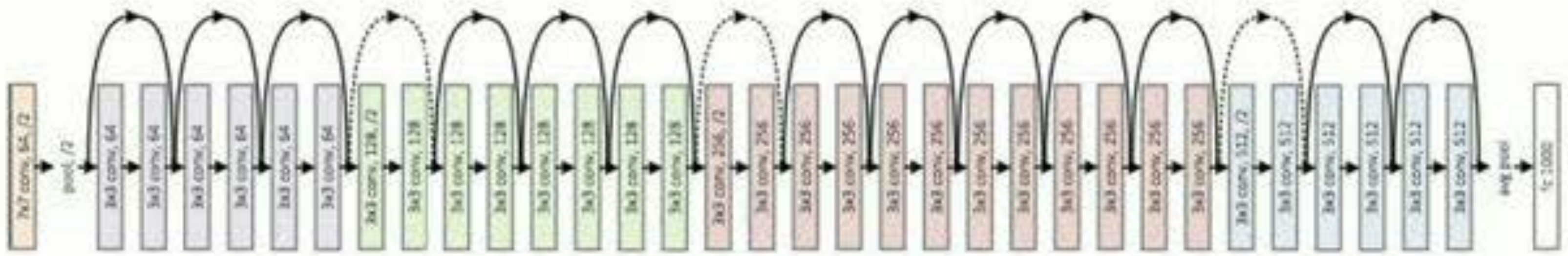
…we can learn from Local Experts!

Example: AlphaGo-Zero

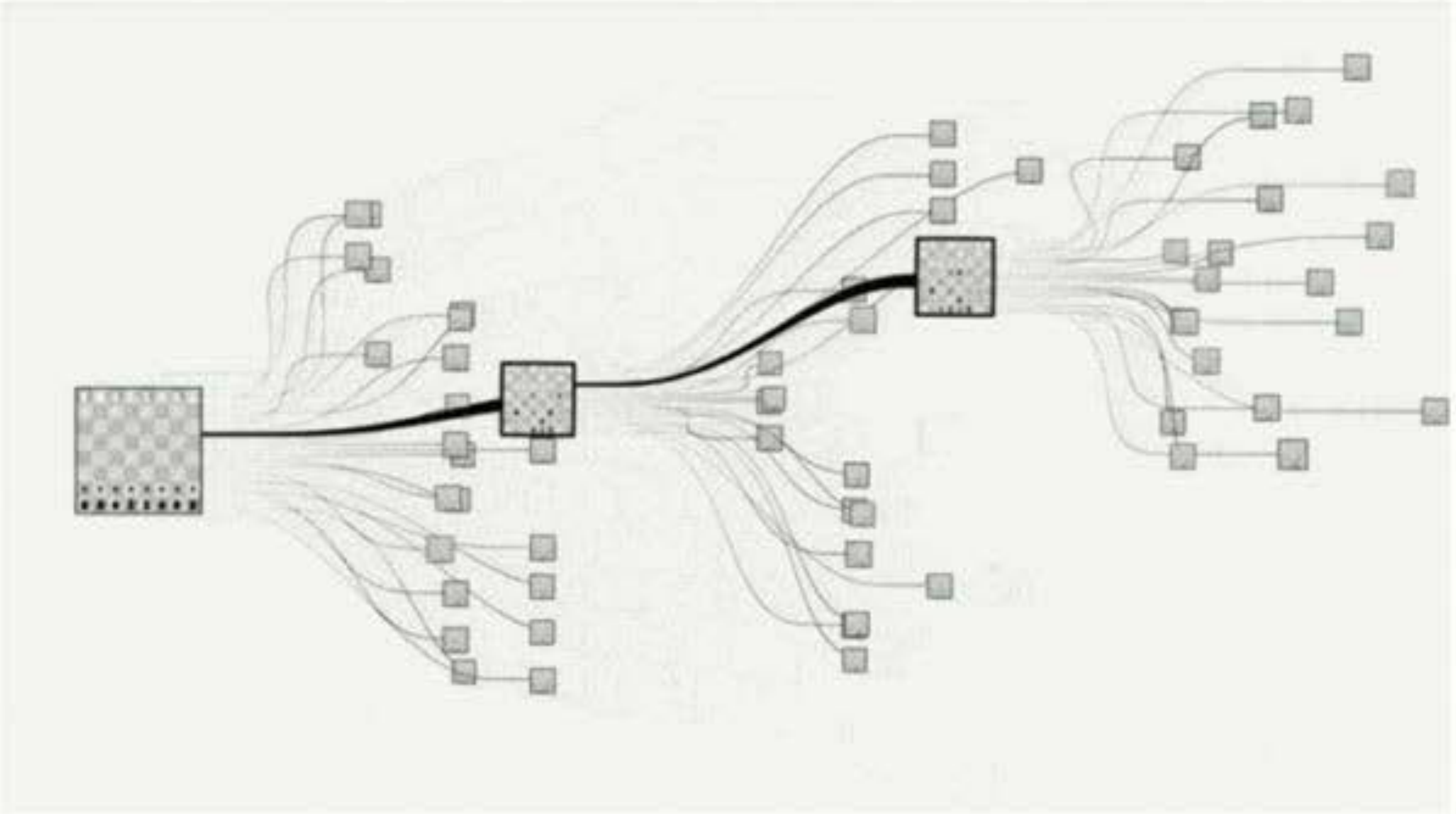
[Silver, et.al, 17, Nature]

Known & **Deterministic** Transition Dynamics

Fast Reactive Policy π



Slow Policy η (MCTS)



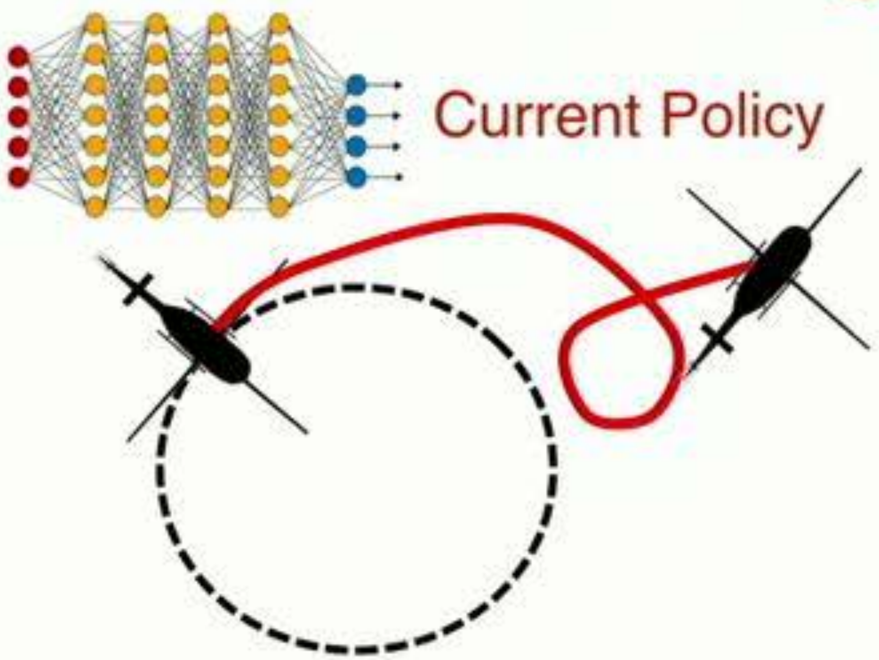
AlphaZero leverages transition dynamics to build local experts

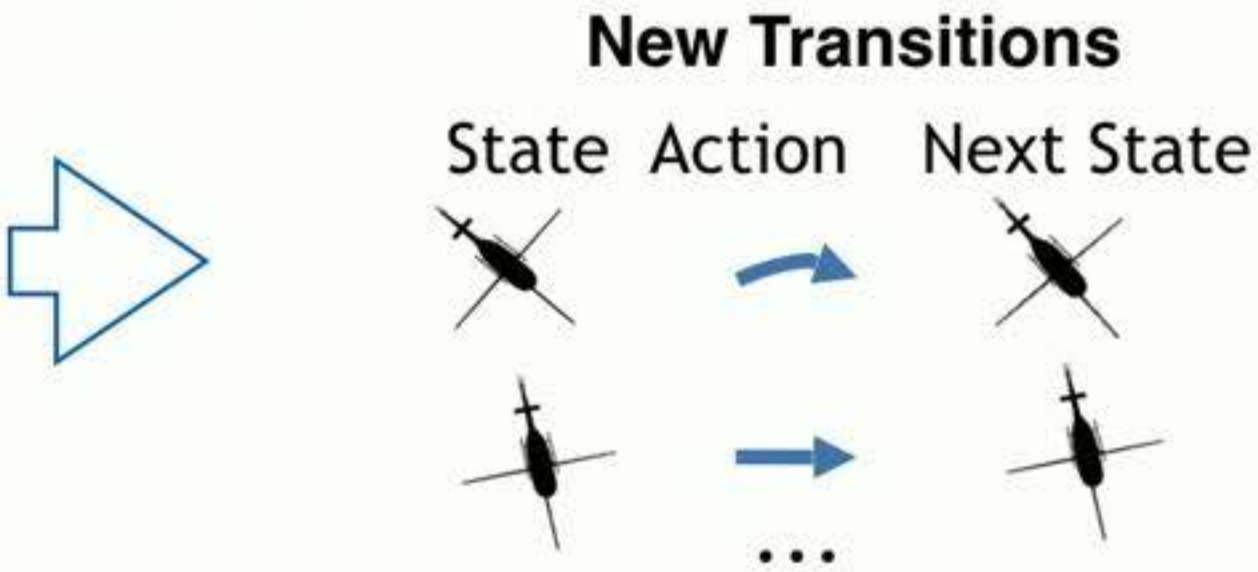
What if we *do not* have any prior knowledge of transition dynamics?

Dual Policy Iteration

[**Sun** et.al., 18, NeurIPS]

Imitating a Locally Optimal Control

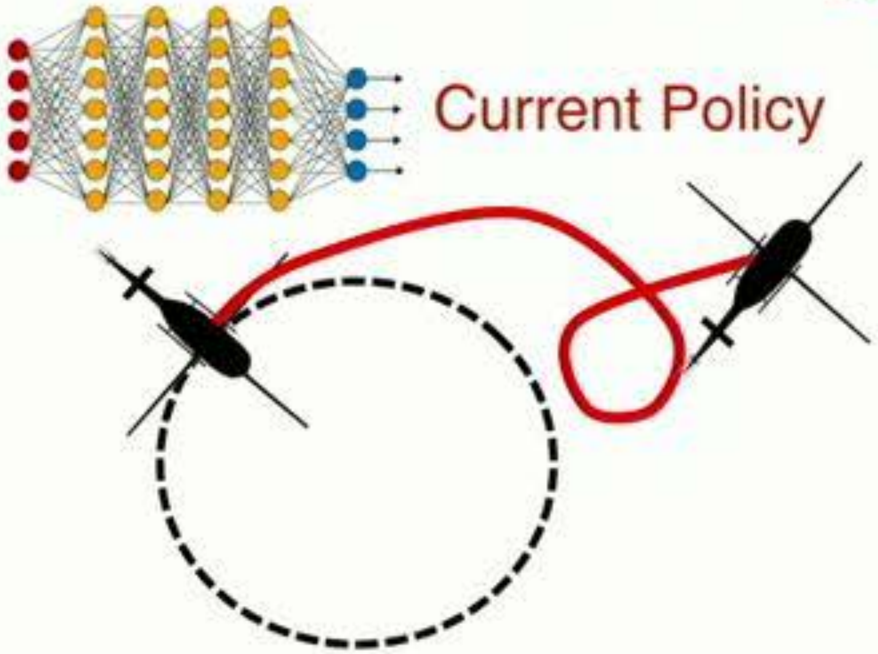




Dual Policy Iteration

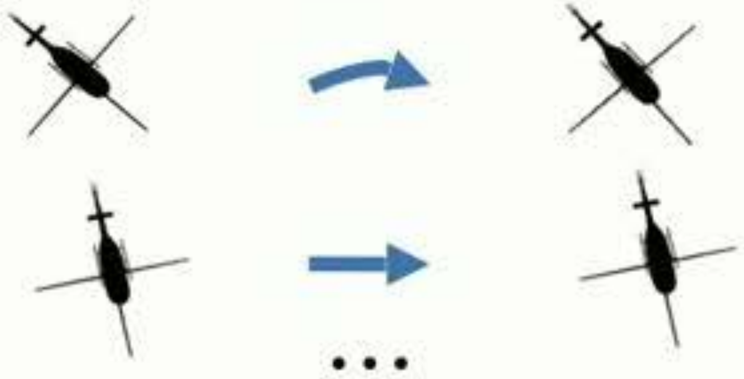
[**Sun** et.al., 18, NeurIPS]

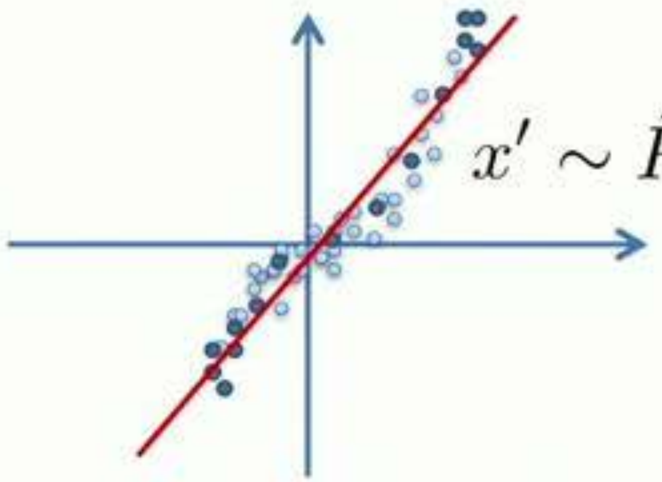
Imitating a Locally Optimal Control



New Transitions

State Action Next State





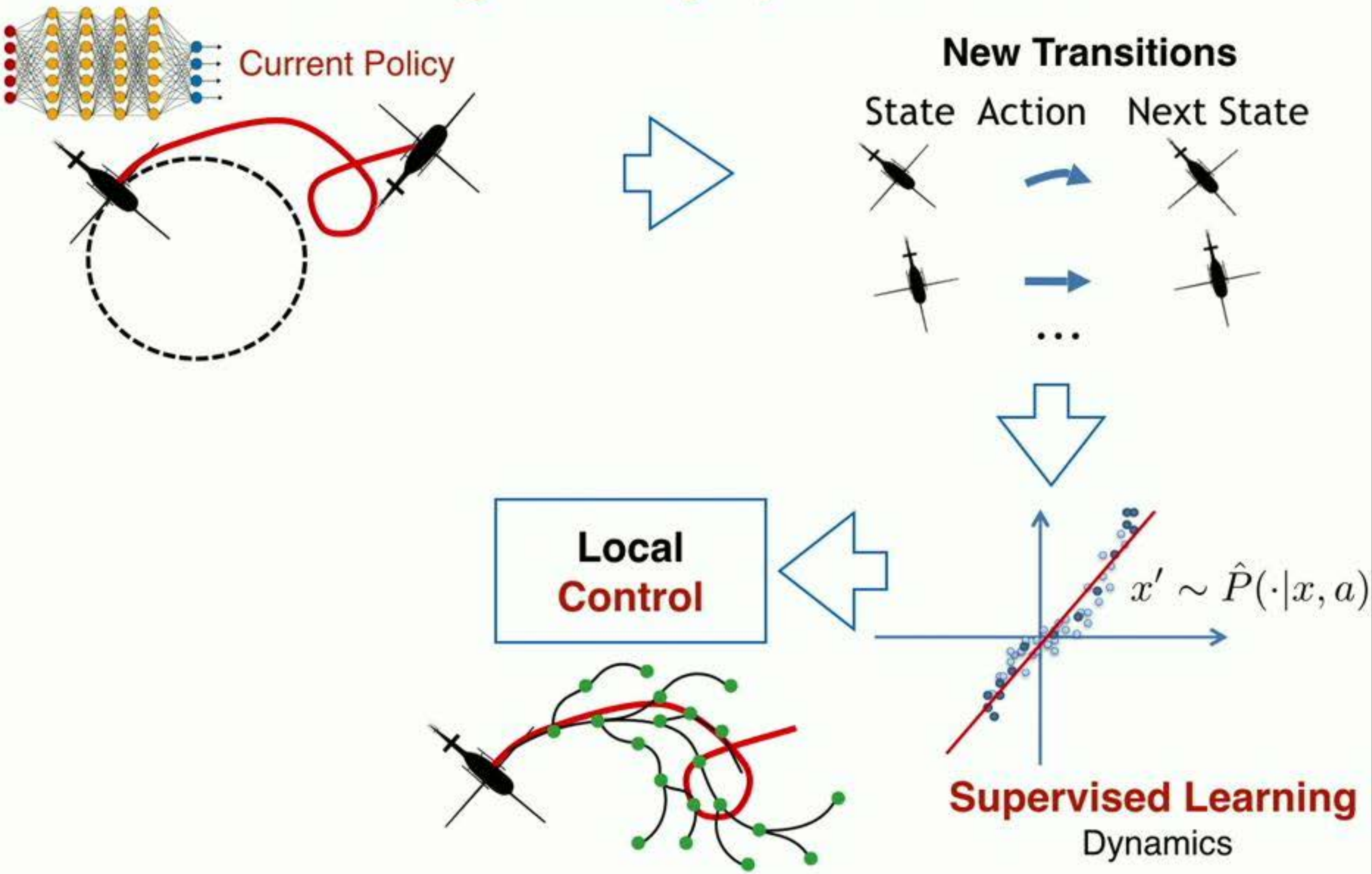
Supervised Learning

Dynamics

Dual Policy Iteration

[**Sun** et.al., 18, NeurIPS]

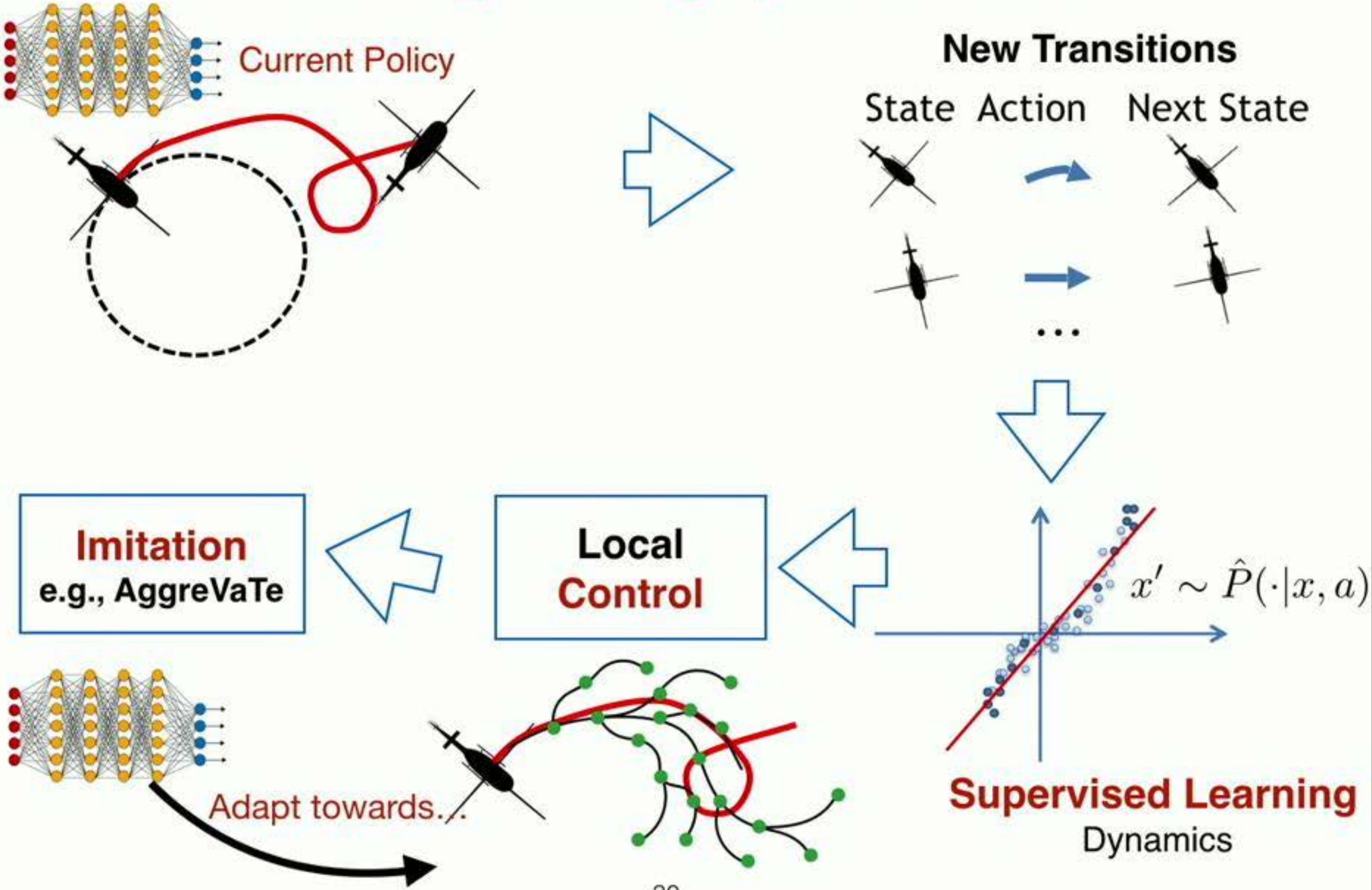
Imitating a Locally Optimal Control



Dual Policy Iteration

[**Sun** et.al., 18, NeurIPS]

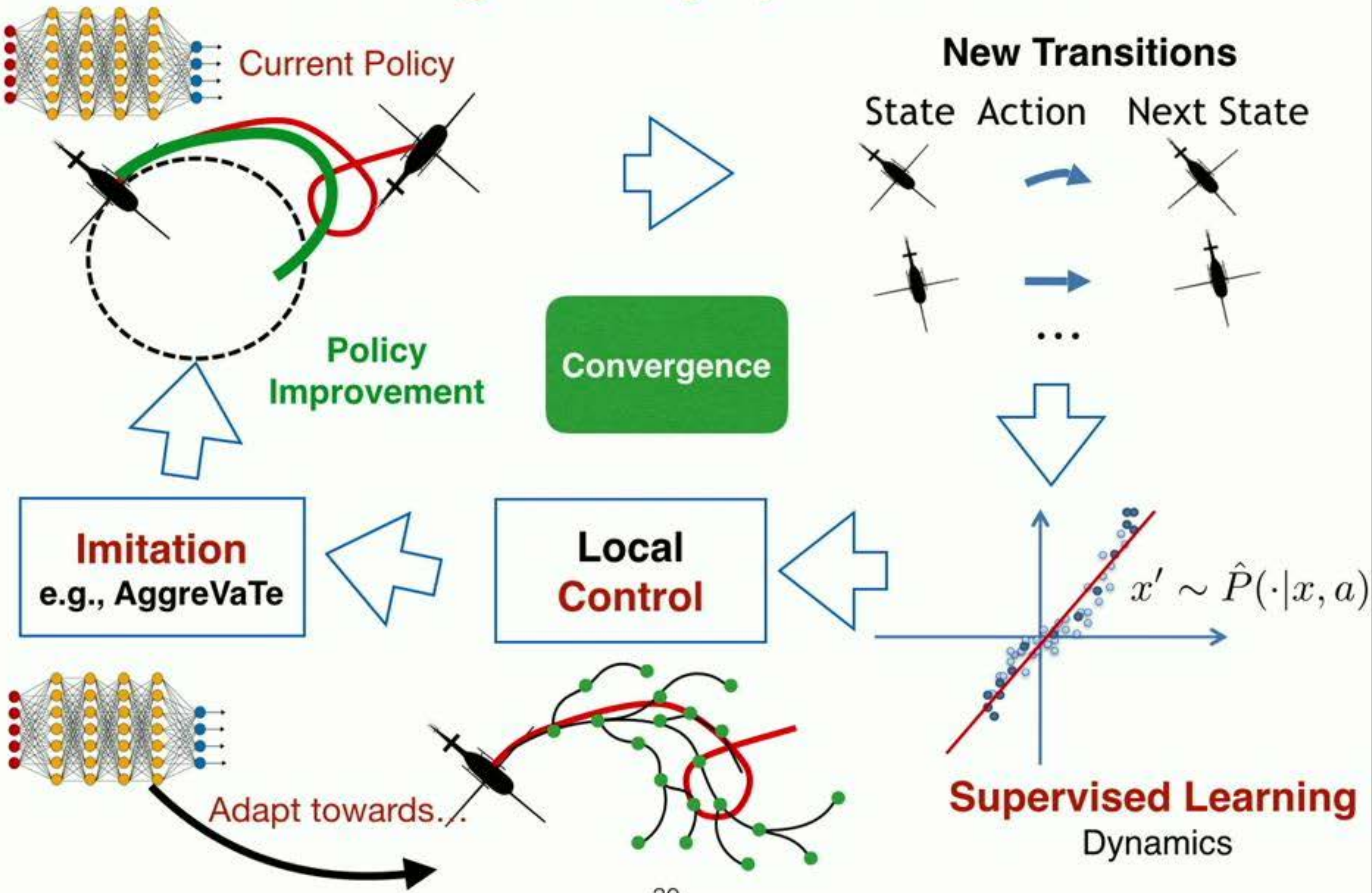
Imitating a Locally Optimal Control



Dual Policy Iteration

[**Sun** et.al., 18, NeurIPS]

Imitating a Locally Optimal Control



Helicopter Funnel

[**Sun** et.al., 18, NeurIPS]

Instantiation 1:
Linear Regressors + iLQR + AggreVaTeD w/ Natural Gradient

iLQR: [Li & Todorov, 05] AggreVaTeD: [Sun, 17, ICML]

Helicopter Funnel

[**Sun** et.al., 18, NeurIPS]

Learned Policy from DPI
(Simulator from Abbeel et.al, 06)

Instantiation 1:
Linear Regressors + iLQR + AggreVaTeD w/ Natural Gradient

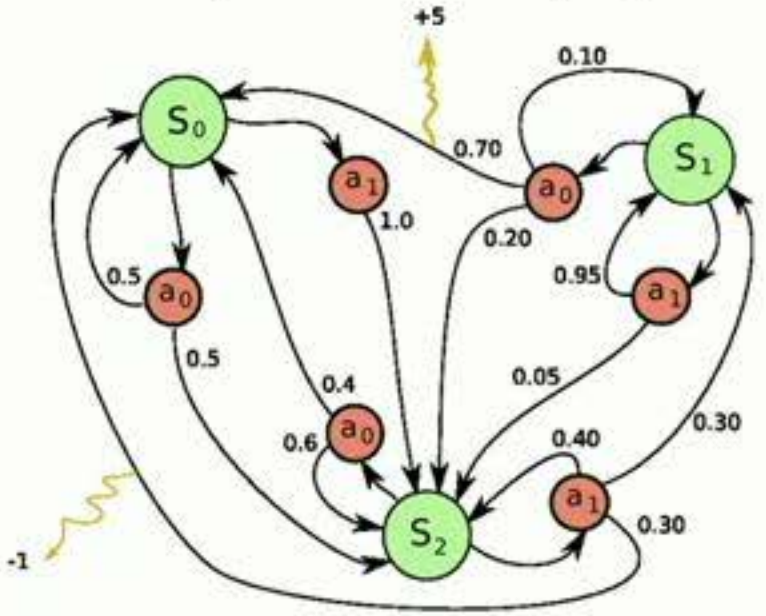
iLQR: [Li & Todorov, 05] AggreVaTeD: [Sun, 17, ICML]

Synthetic Discrete MDPs

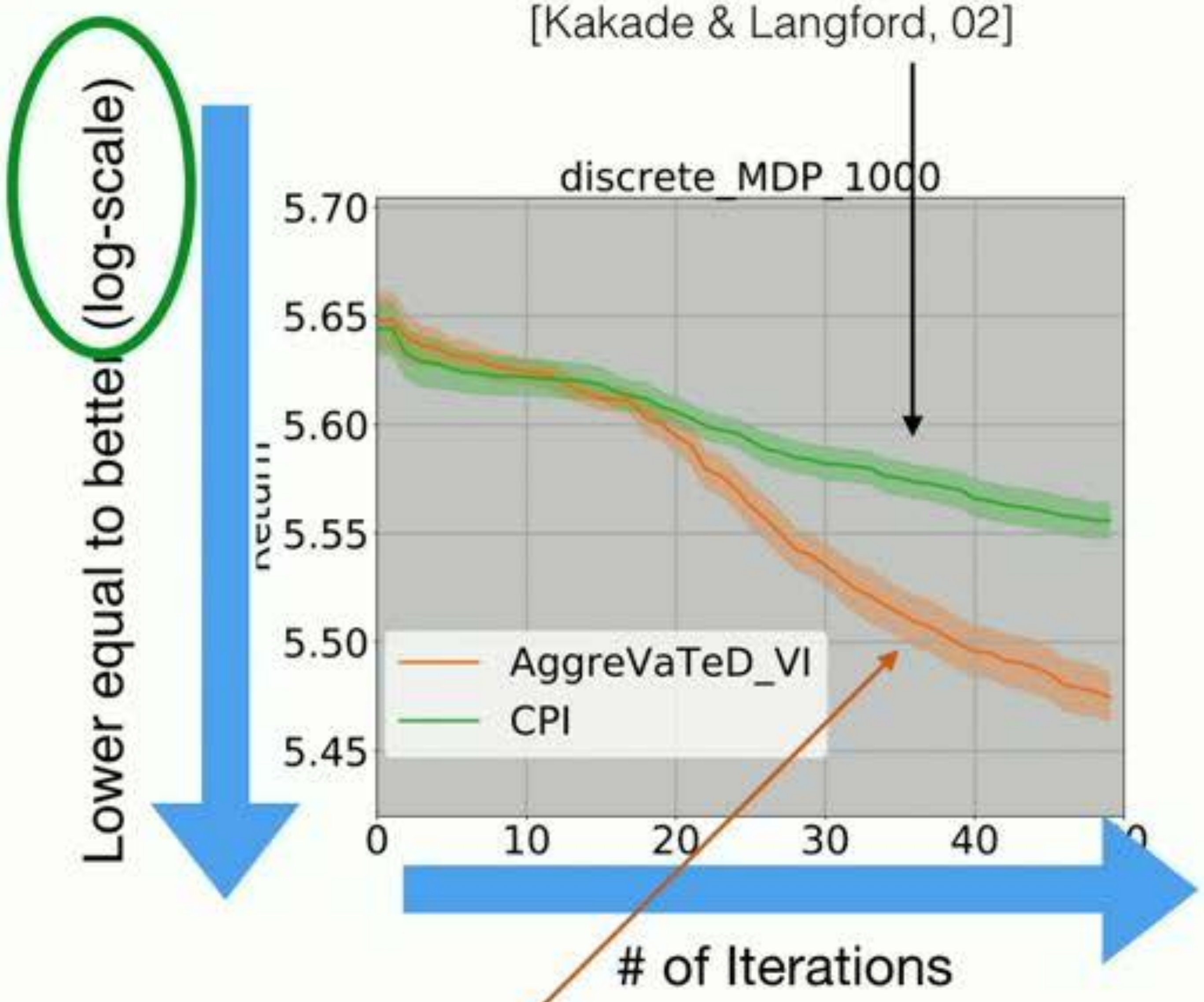
[**Sun** et.al., 18, NeurIPS]

Randomly Generated
Discrete MDPs

[Archibald et.al., 95]



Conservative Policy Iteration
[Kakade & Langford, 02]



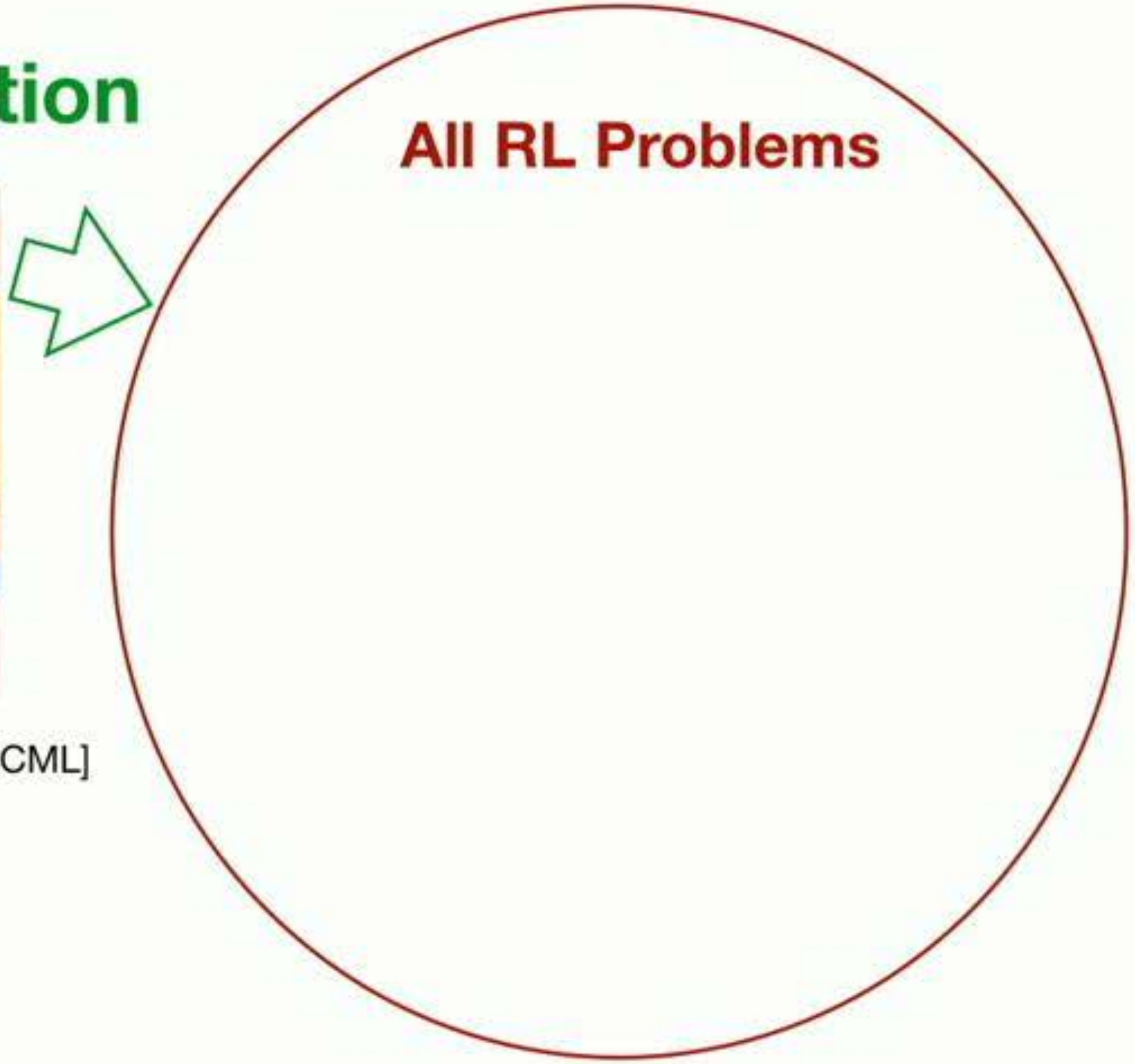
Instantiation 2:
Maximum Likelihood Estimation + Value Iteration + AggreVaTeD

AggreVaTeD: [Sun, 17, ICML]

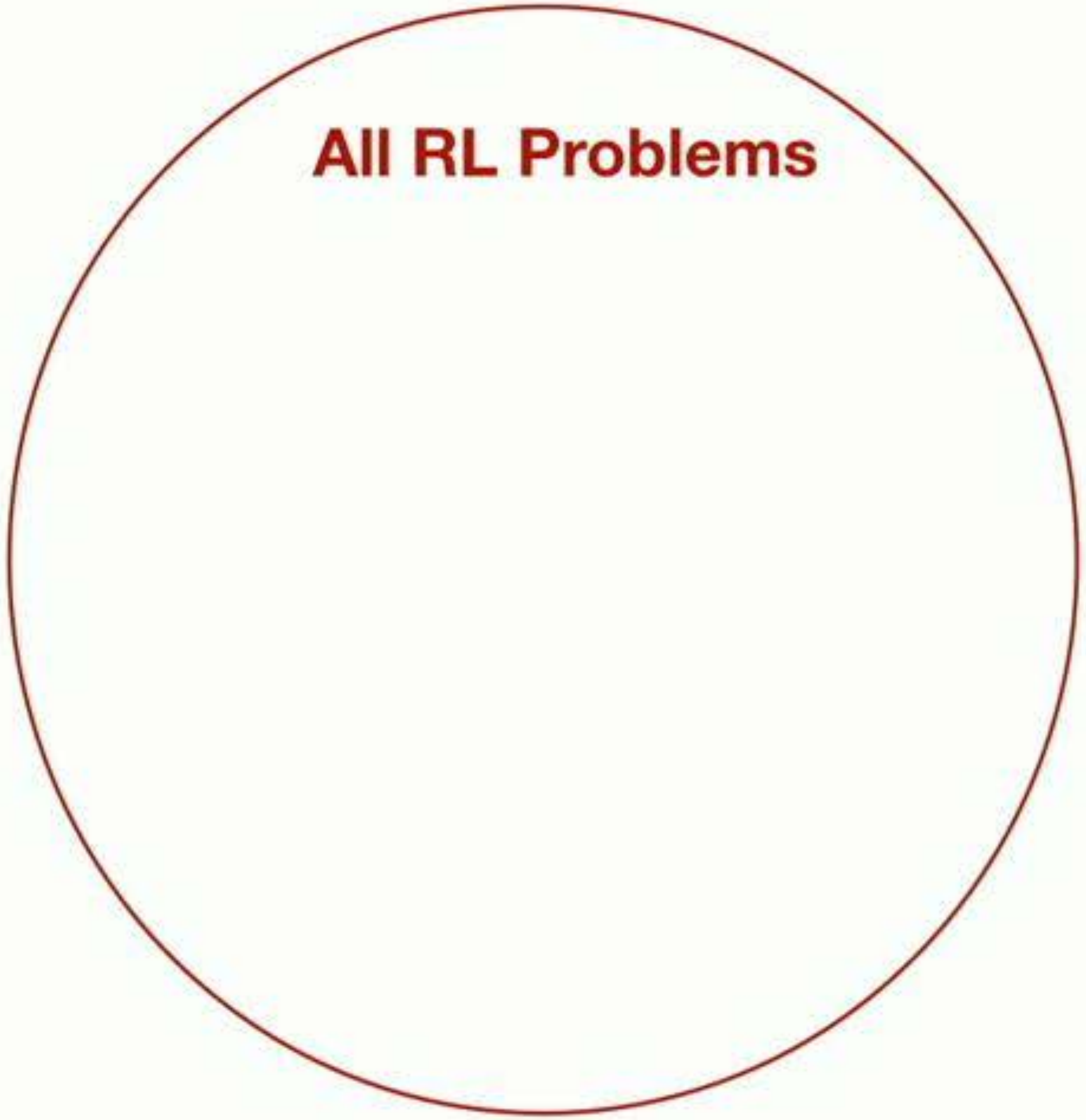
Generalization & Sample Efficiency via…

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]
[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]

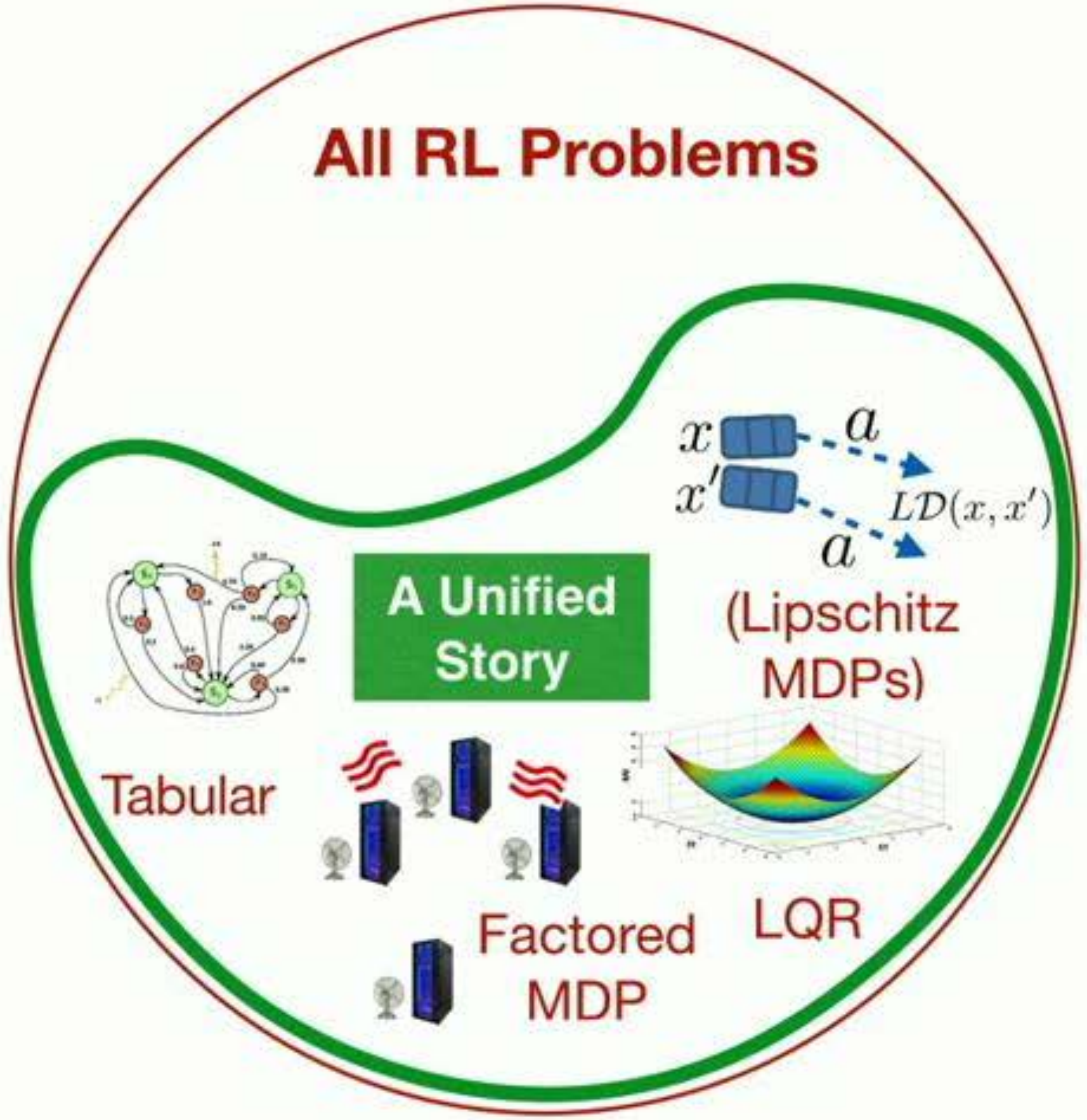


Generalization & Sample Efficiency via…



Generalization & Sample Efficiency via...

- Why Model-Based RL?
- A Unified Measure

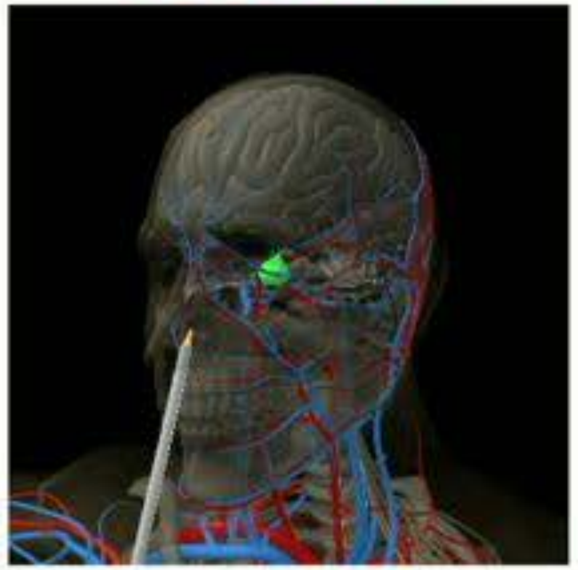


2. Exploiting Structures

[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Modeling Dynamics

Known



[Sun et.al, ISRR 13]

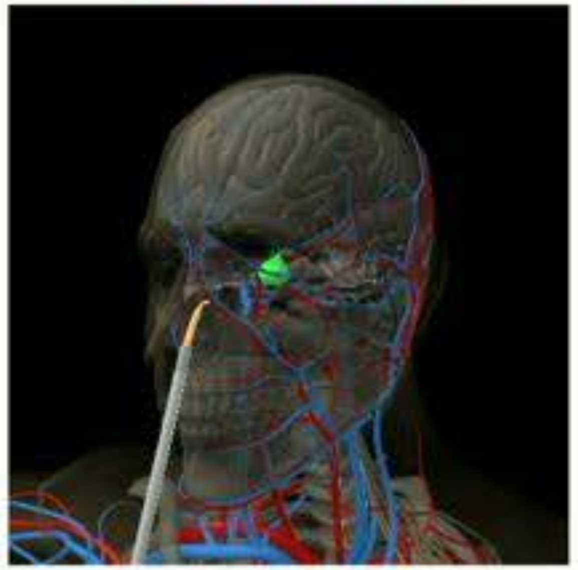
Control

e.g., iterative LQR

[Li & Todorov 03]

Modeling Dynamics

Known



[**Sun** et.al, ISRR 13]

Control

e.g., iterative LQR

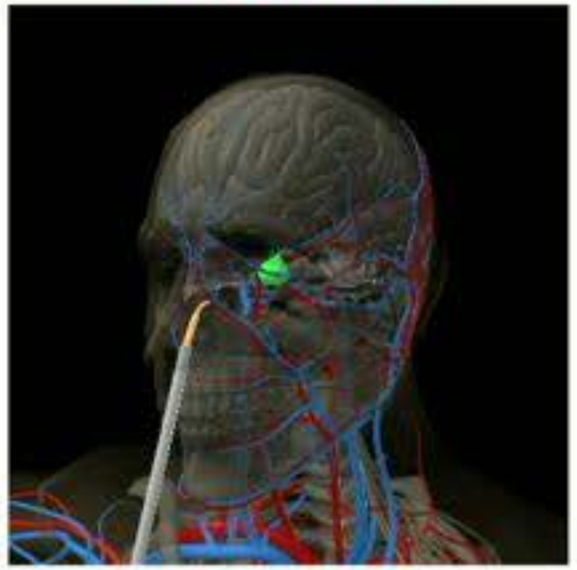
[Li & Todorov 03]

Learned

[Williams et.al, 17, ICRA]

Modeling Dynamics

Known



[Sun et.al, ISRR 13]

Control

e.g., iterative LQR

[Li & Todorov 03]

Learned

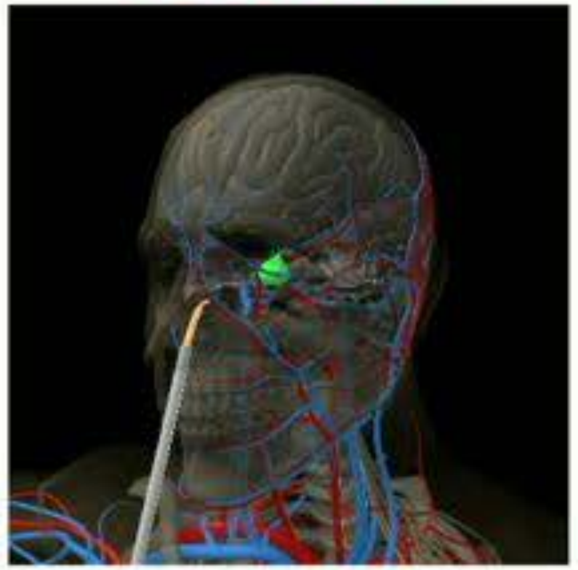
[Williams et.al, 17, ICRA]

Model-Based RL

$$\hat{P}(\cdot|x,a) \approx P^{\star}(\cdot|x,a)$$

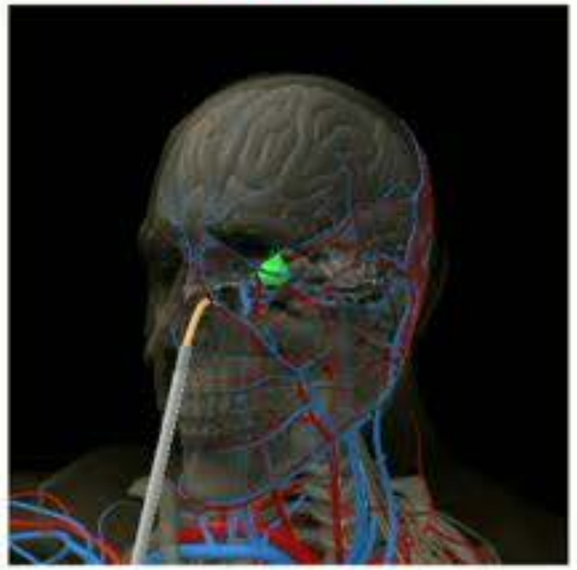
Approximator   Real Transition

Modeling Dynamics

Known	Learned	Ignored
		
[**Sun** et.al, ISRR 13]	[Williams et.al, 17, ICRA]	
Control	**Model-Based RL**	Directly learn policy **Model-Free RL**
e.g., iterative LQR [Li & Todorov 03]	$\hat{P}(\cdot \vert x, a) \approx P^{\star}(\cdot \vert x, a)$ Approximator Real Transition	e.g., Q-Learning [Watkins & Dayan, 92]

Modeling Dynamics

Known



[Sun et.al, ISRR 13]

Control

e.g., iterative LQR

[Li & Todorov 03]

Learned

[Williams et.al, 17, ICRA]

Model-Based RL

$$\hat{P}(\cdot|x,a) \approx P^{\star}(\cdot|x,a)$$

Approximator Real Transition

Ignored

Directly learn policy

Model-Free RL

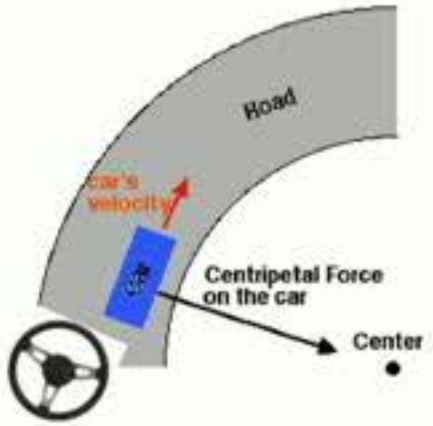
e.g., Q-Learning

[Watkins & Dayan, 92]

Setup of Model-Based RL

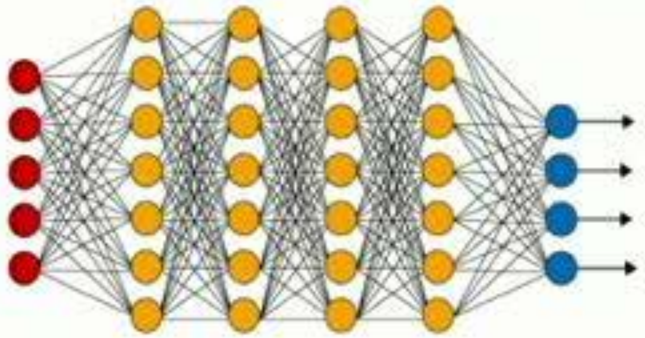
$$x' \sim P^\star(\cdot|x,a)$$

Real Transition Dynamics



Setup of Model-Based RL

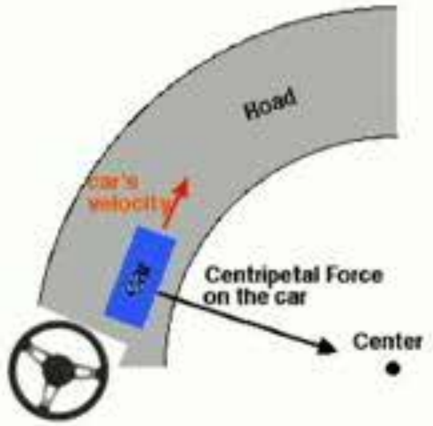
Function Approximators

(x, a)  $P(\cdot|x, a)$

$$\mathcal{P} = \{P : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})\}$$

$$x' \sim P^\star(\cdot|x, a)$$

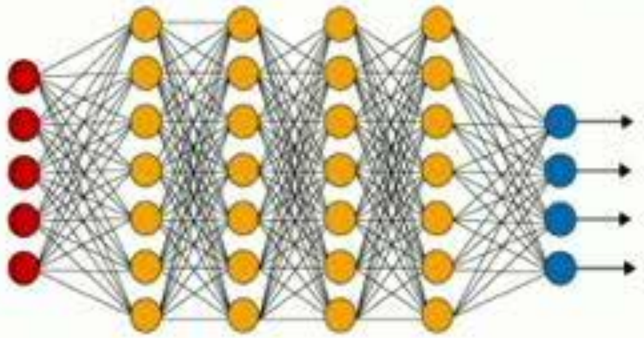
Real Transition Dynamics



Setup of Model-Based RL

Function Approximators

(x, a)



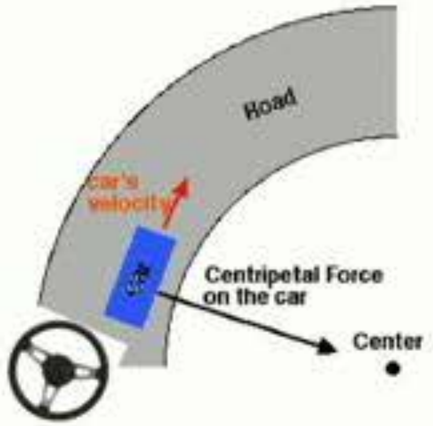
$P(\cdot|x, a)$

$$\mathcal{P} = \{P : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})\}$$

Realizability: $P^\star \in \mathcal{P}$

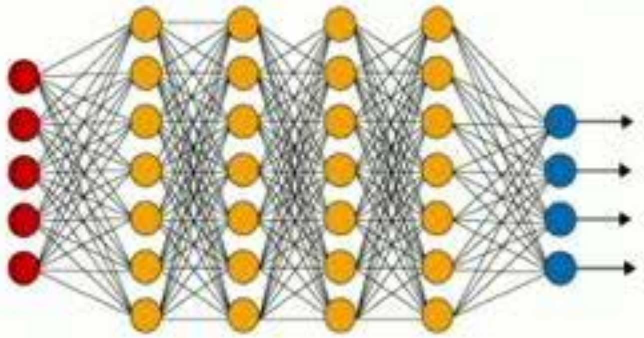
$$x' \sim P^\star(\cdot|x, a)$$

Real Transition Dynamics



Setup of Model-Based RL

Function Approximators

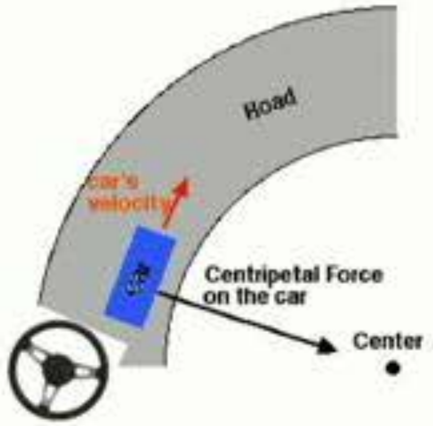
(x, a)  $P(\cdot|x, a)$

$$\mathcal{P} = \{P : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})\}$$

Realizability: $P^\star \in \mathcal{P}$

$$x' \sim P^\star(\cdot|x, a)$$

Real Transition Dynamics



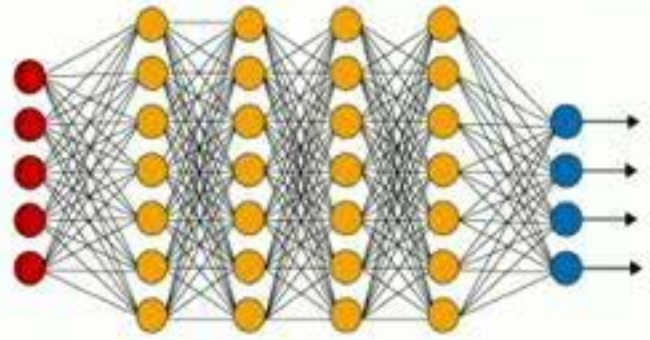
Optimal Planner (OP)

$$OP(P, r) \Rightarrow \pi_P$$

Setup of Model-Based RL

Function Approximators

(x, a)



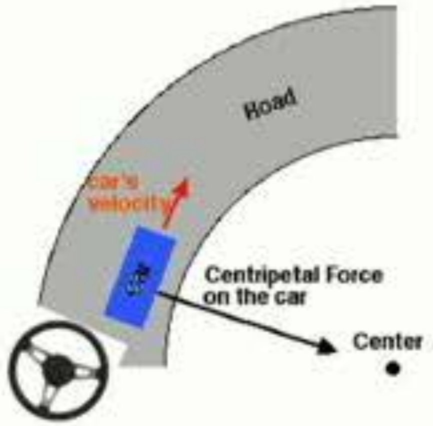
$P(\cdot|x, a)$

$$\mathcal{P} = \{P : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})\}$$

Realizability: $P^\star \in \mathcal{P}$

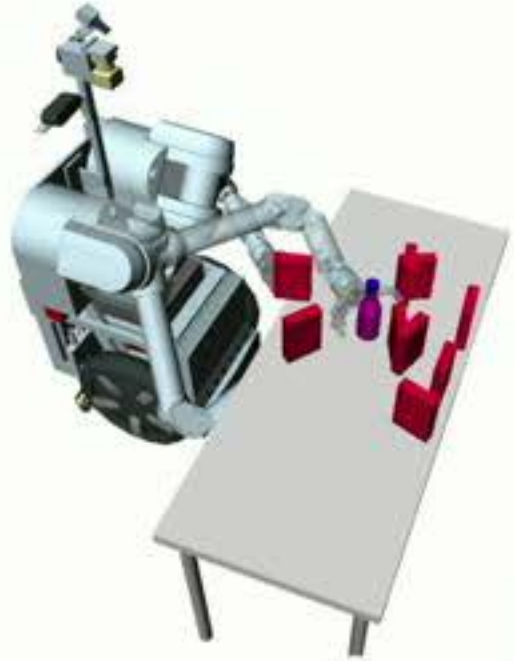
$$x' \sim P^\star(\cdot|x, a)$$

Real Transition Dynamics

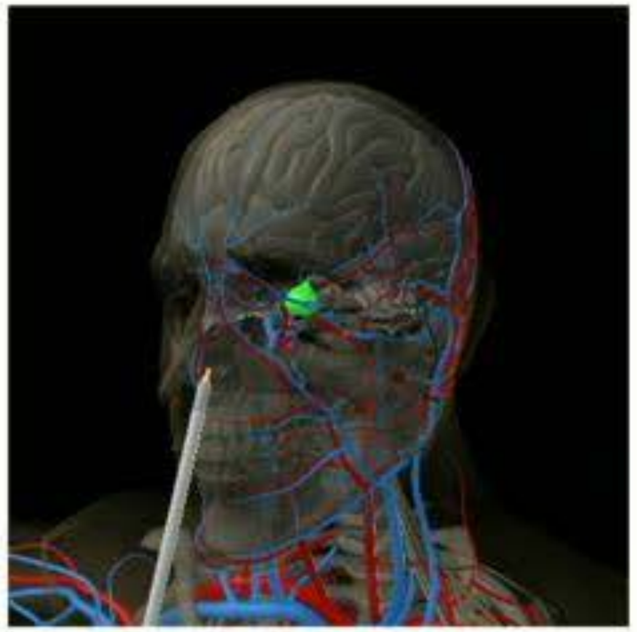


Optimal Planner (OP)

$$OP(P, r) \Rightarrow \pi_P$$



[Zucker et.al, IJRR 13]



[Sun et.al, ISRR 13]

e.g., iLQR [Li & Todorov 03]
CHOMP [Ratliff et.al, 09]
SE-LQR [**Sun** et.al, 16, TASE]

Why Model-Based RL?

Debate: Model-Based or Model-Free

Iterative Learning Control
(e.g., An & Atkeson & Hollerbach 88, Abbeel 06)

Nonparametric Model-based RL
(e.g., Atkeson 98, Deisenroth et.al., 11)

Guided Policy Search
(e.g.,Levine & Abbeel 16)

Dual Policy Iteration
[**Sun** et.al, 18]

...

Why Model-Based RL?

Debate: Model-Based or Model-Free

Model-Based is often more sample **efficient** than Model-Free **in practice**…

Iterative Learning Control
(e.g., An & Atkeson & Hollerbach 88, Abbeel 06)

Nonparametric Model-based RL
(e.g., Atkeson 98, Deisenroth et.al., 11)

Guided Policy Search
(e.g.,Levine & Abbeel 16)

Dual Policy Iteration
[**Sun** et.al, 18]

…

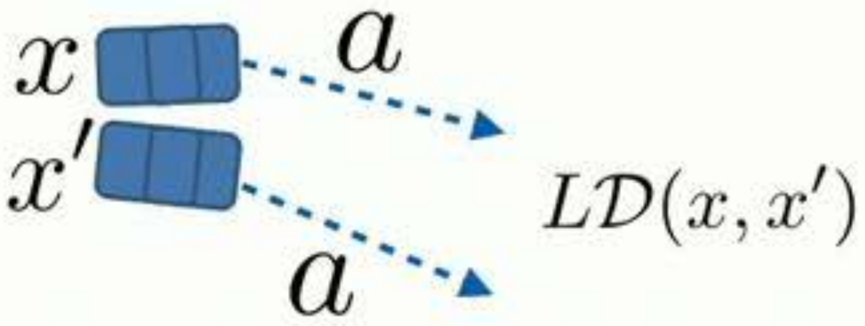
In Theory?

There exists MDPs (e.g., Factored MDPs), s.t., to learn near optimal policy,

Model-Based RL:		Any Model-Free RL:
Polynomial Sample Complexity	**VS**	$\Omega(\exp(H))$

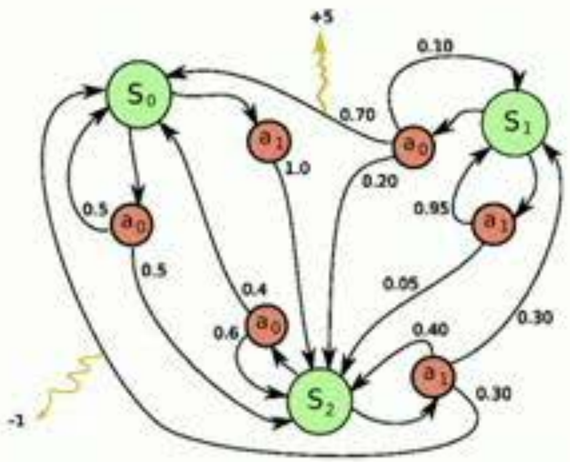
Model-based Reinforcement Learning in Contextual Decision Processes
Sun, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18.

We have been exploiting the structures of models, BUT…



Lipschitz Continuous MDPs

[Kearn, Langford, Kakade, 03]

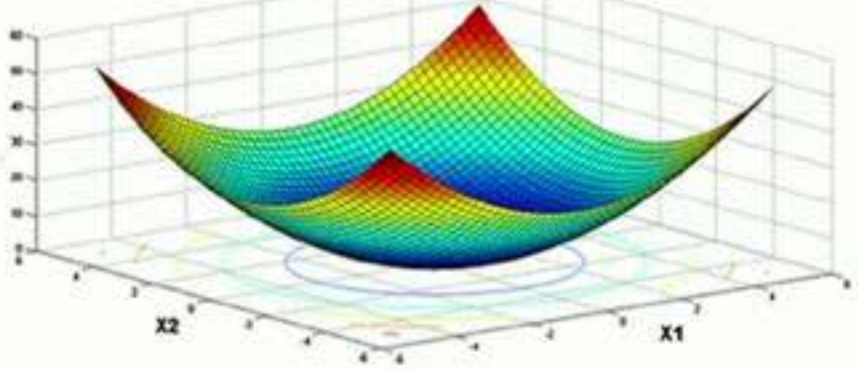


Small Tabular MDP

[Kearn & Singh, 02]

Factored MDPs

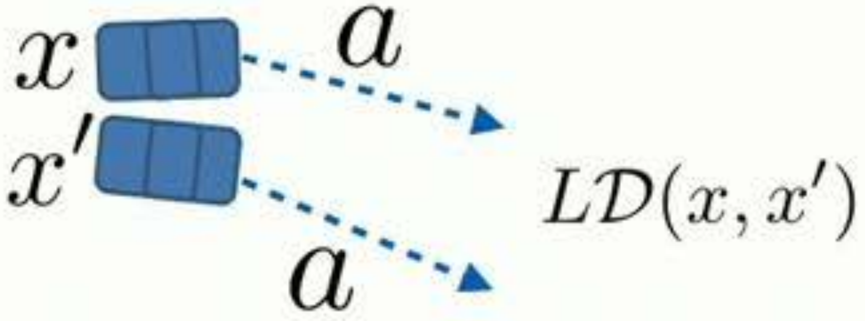
[Guestrin et.al, 03; Osband & Van Roy,13]



Linear Quadratic Regulator (LQR)

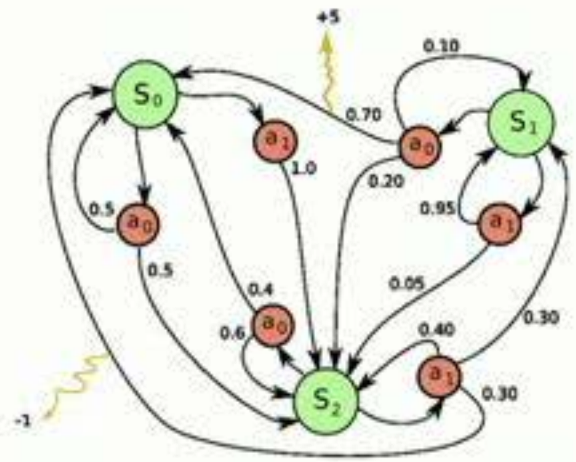
[Dean et.al, 18]

We have been exploiting the structures of models, BUT…



Lipschitz Continuous MDPs

[Kearn, Langford, Kakade, 03]



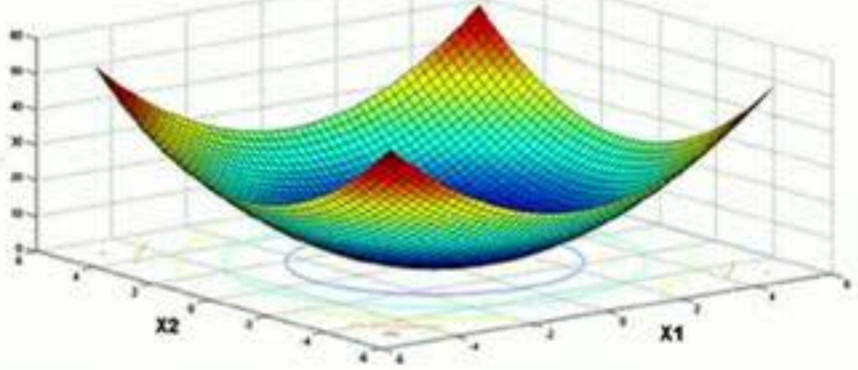
Small Tabular MDP

[Kearn & Singh, 02]

A Unified Algorithm?

Factored MDPs

[Guestrin et.al, 03; Osband & Van Roy,13]



Linear Quadratic Regulator (LQR)

[Dean et.al, 18]

Distinguish Two Distributions:

Integral Probability Metric (IPM) [Muller et.al, 97]

Distinguish two distributions P, Q

Distinguish Two Distributions:

Integral Probability Metric (IPM) [Muller et.al, 97]

Distinguish two distributions P, Q

Real bedroom images
[LSUN dataset]

Imaginary samples from a generative model
[e.g., Wasserstein GAN,17]

Distinguish Two Distributions:

Integral Probability Metric (IPM) [Muller et.al, 97]

Distinguish two distributions P, Q

Real bedroom images
[LSUN dataset]

Imaginary samples from a generative model
[e.g., Wasserstein GAN,17]

Discriminators $$\max_{f \in \mathcal{F}} [\mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{x \sim Q} f(x)]$$

Distinguish Two Distributions:

Integral Probability Metric (IPM) [Muller et.al, 97]

Distinguish two distributions P, Q

Real bedroom images
[LSUN dataset]

Imaginary samples from a generative model
[e.g., Wasserstein GAN,17]

Discriminators $$\max_{f \in \mathcal{F}} [\mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{x \sim Q} f(x)]$$

$$\mathcal{F} \triangleq \{f : \|f\|_\infty \leq 1\} \Rightarrow \|P - Q\|_1$$ Total Variation

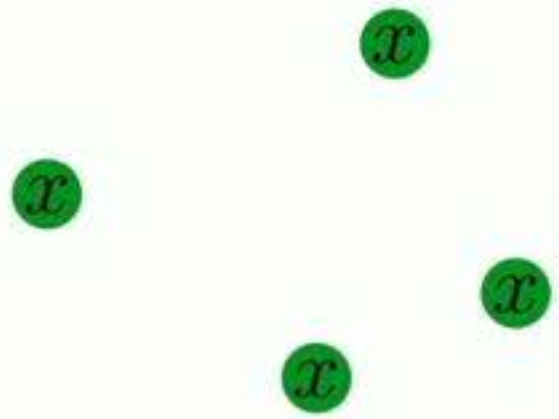
$$\mathcal{F} \triangleq \{f : \|f\|_L \leq 1\} \Rightarrow$$ Wasserstein Distance

Distinguish a Candidate from the Real

Candiate: $P(\cdot|x,a) \overset{?}{=}$ Real: $P^\star(\cdot|x,a)$

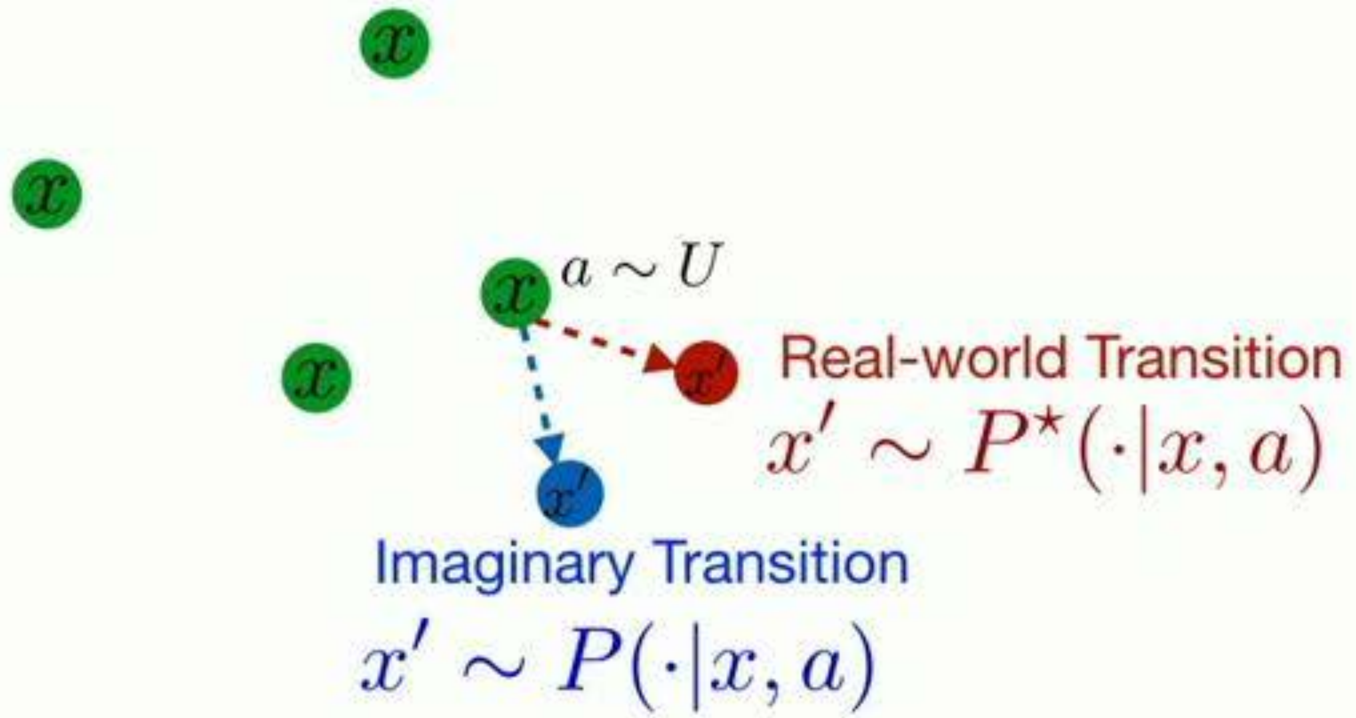
Distinguish a Candidate from the Real

Candiate: $P(\cdot|x, a)$ $\overset{?}{=}$ Real: $P^{\star}(\cdot|x, a)$



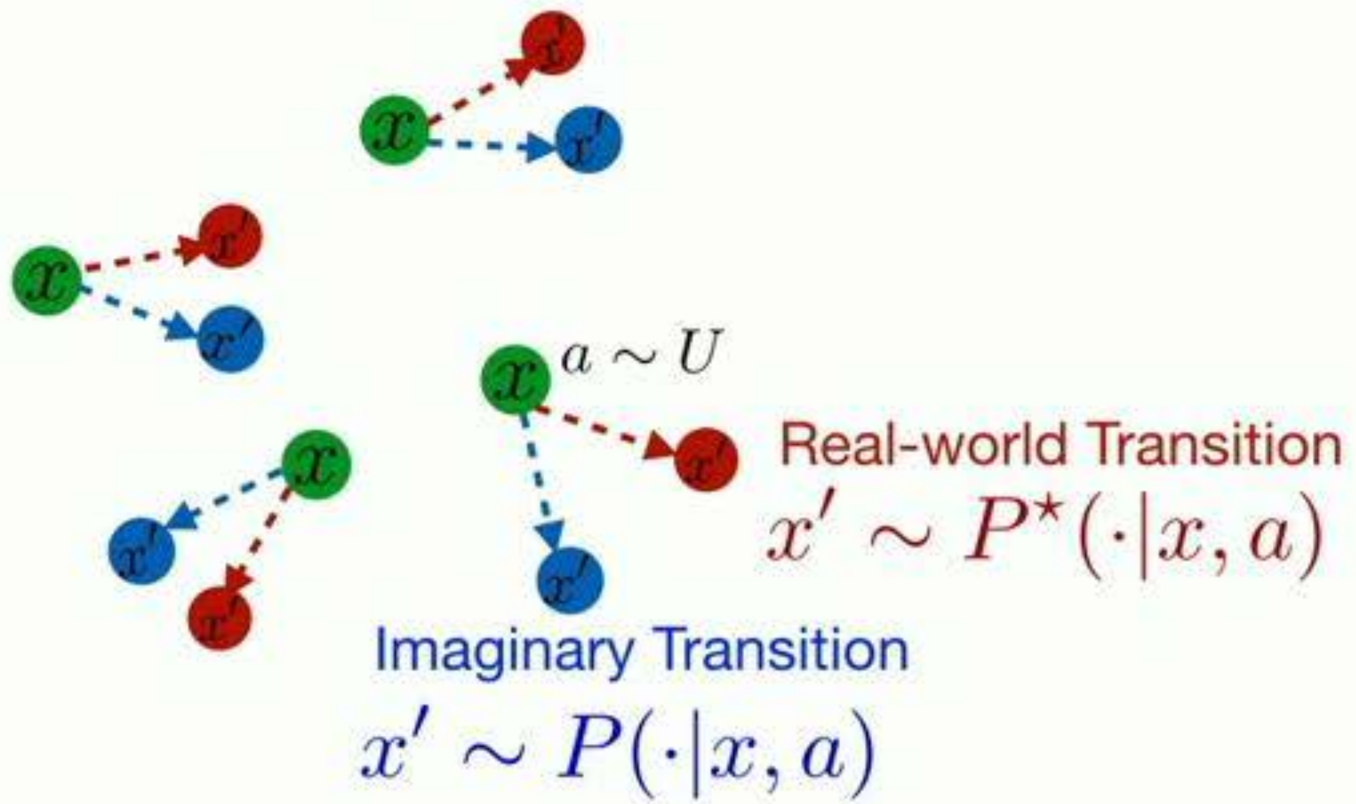
Distinguish a Candidate from the Real

Candiate: $P(\cdot|x, a)$ $\stackrel{?}{=}$ Real: $P^{\star}(\cdot|x, a)$



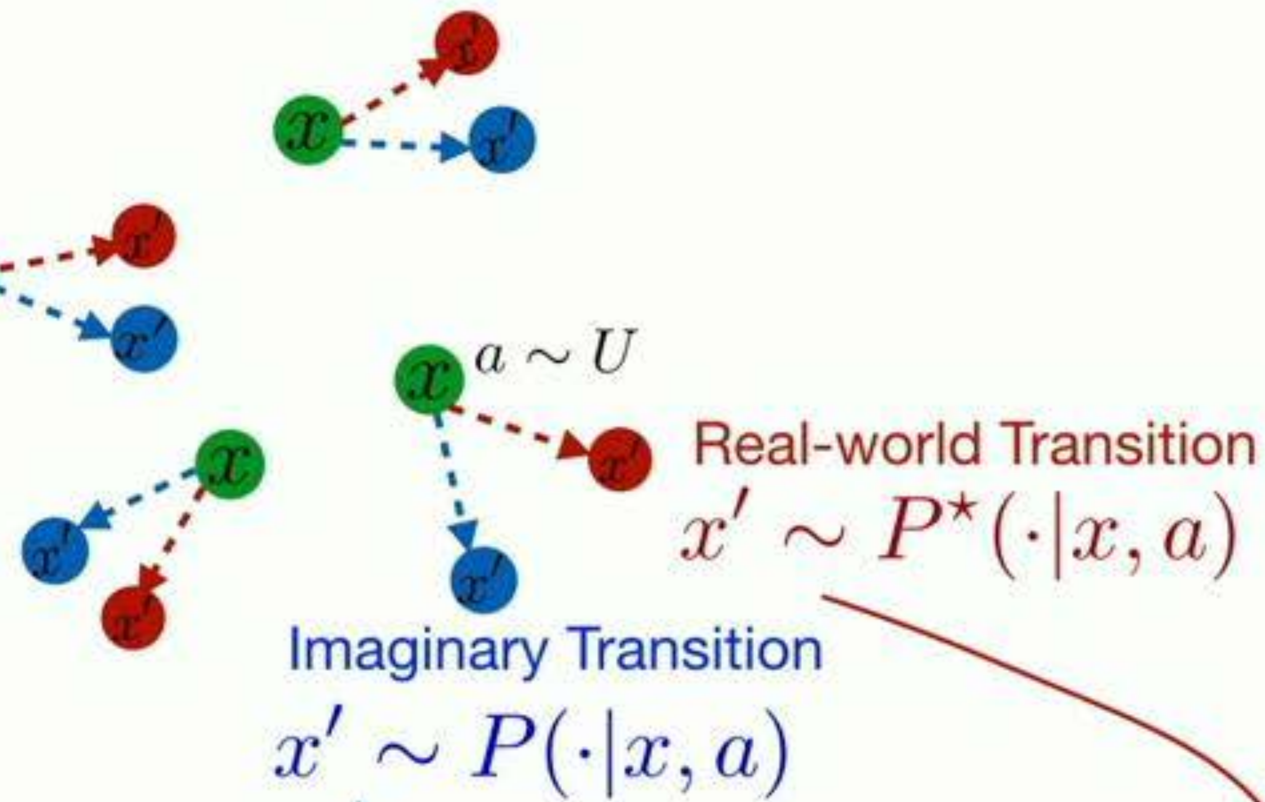
Distinguish a Candidate from the Real

Candiate: $P(\cdot|x,a)$ $\stackrel{?}{=}$ Real: $P^\star(\cdot|x,a)$



Distinguish a Candidate from the Real

Candiate: $P(\cdot|x,a)$ $\stackrel{?}{=}$ Real: $P^{\star}(\cdot|x,a)$



$$\max_{f \in \mathcal{F}} \mathbb{E}_{x \sim \pi, a \sim U}\left[\mathbb{E}_{x' \sim P} f(x,a,x') - \mathbb{E}_{x' \sim P^{\star}} f(x,a,x')\right]$$

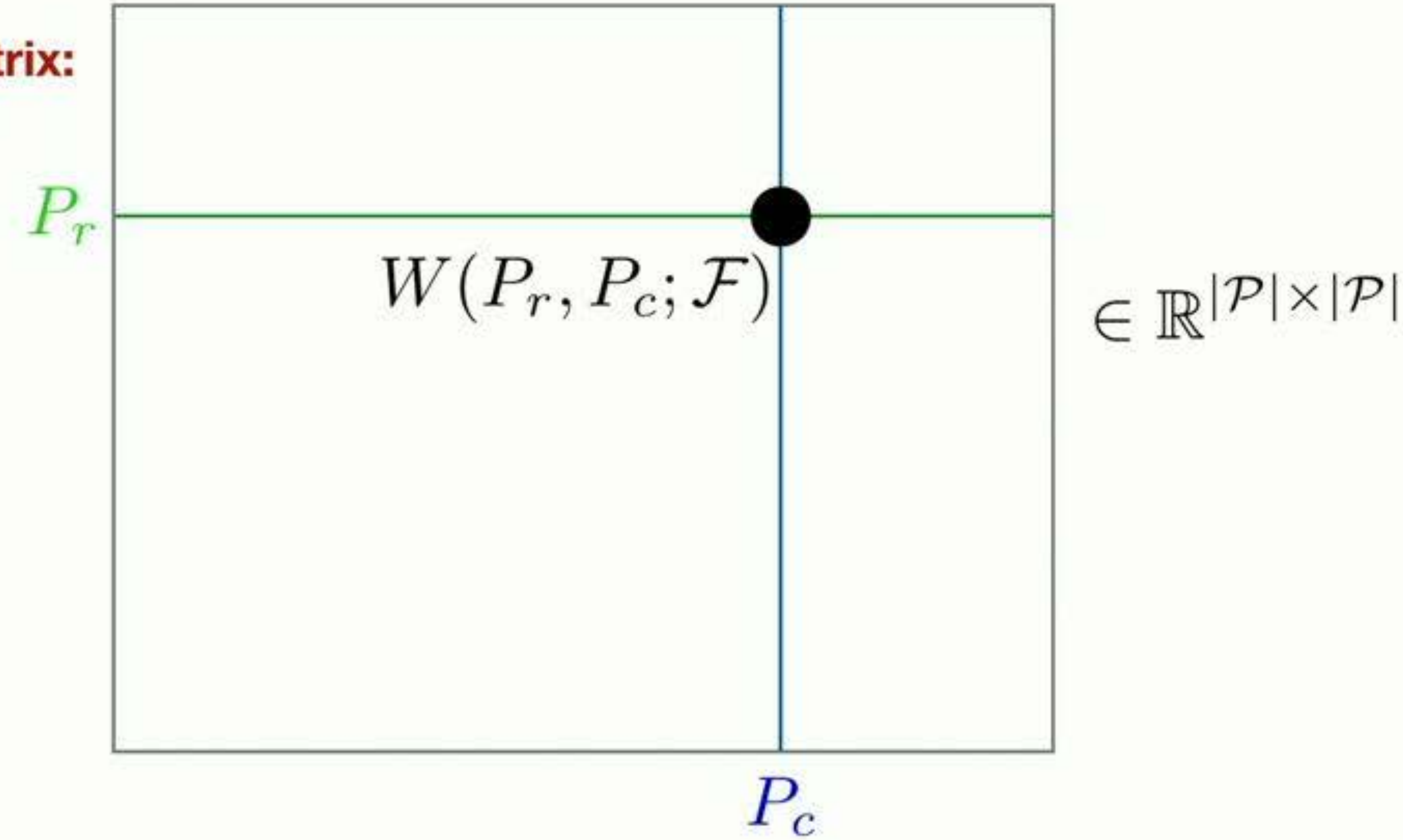
Model Rank

Misfit Matrix:

$$\in \mathbb{R}^{|\mathcal{P}| \times |\mathcal{P}|}$$

Model Rank

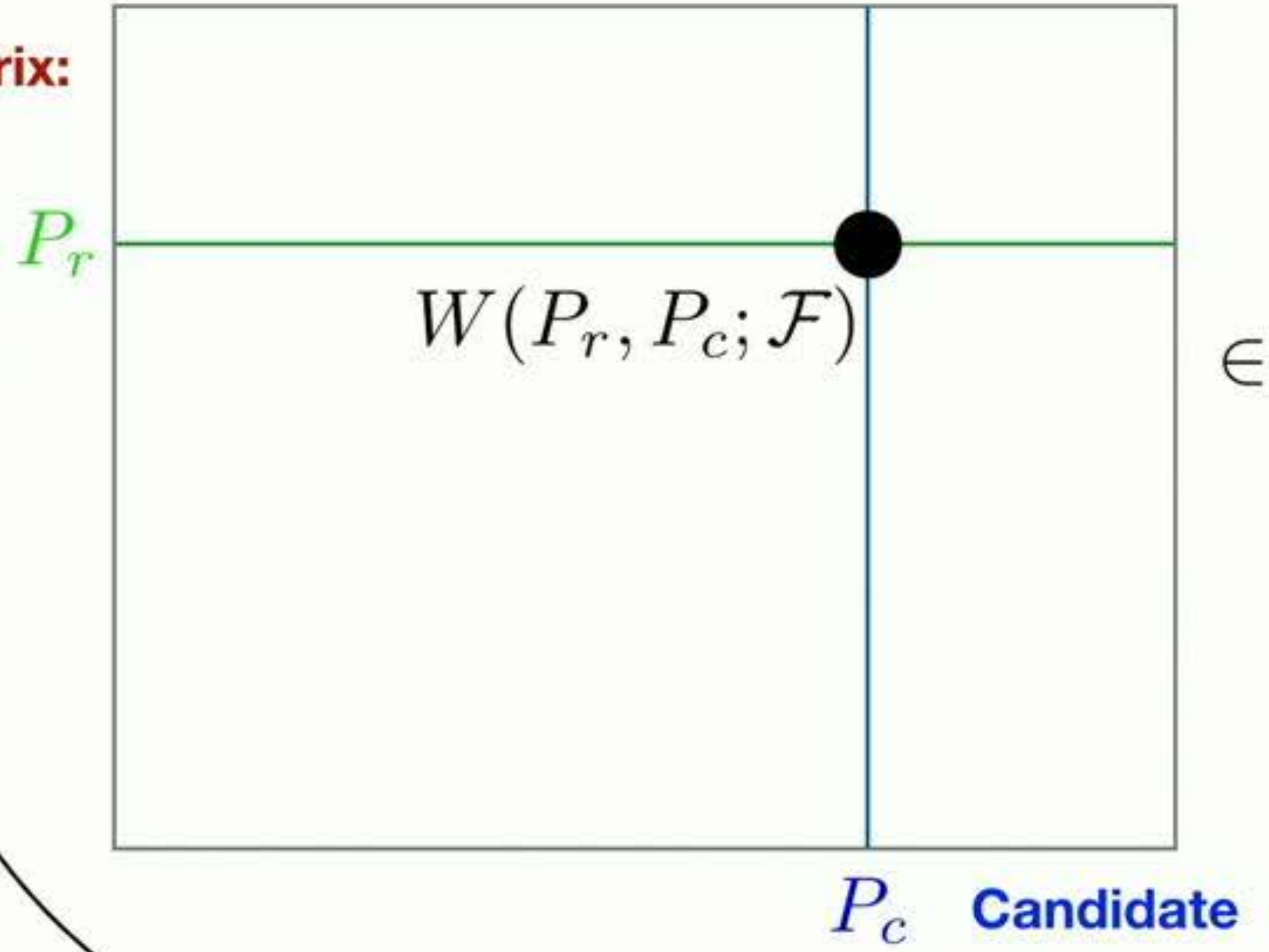
Misfit Matrix:



Model Rank

Misfit Matrix:

Provides **Conditional** State-action Distribution

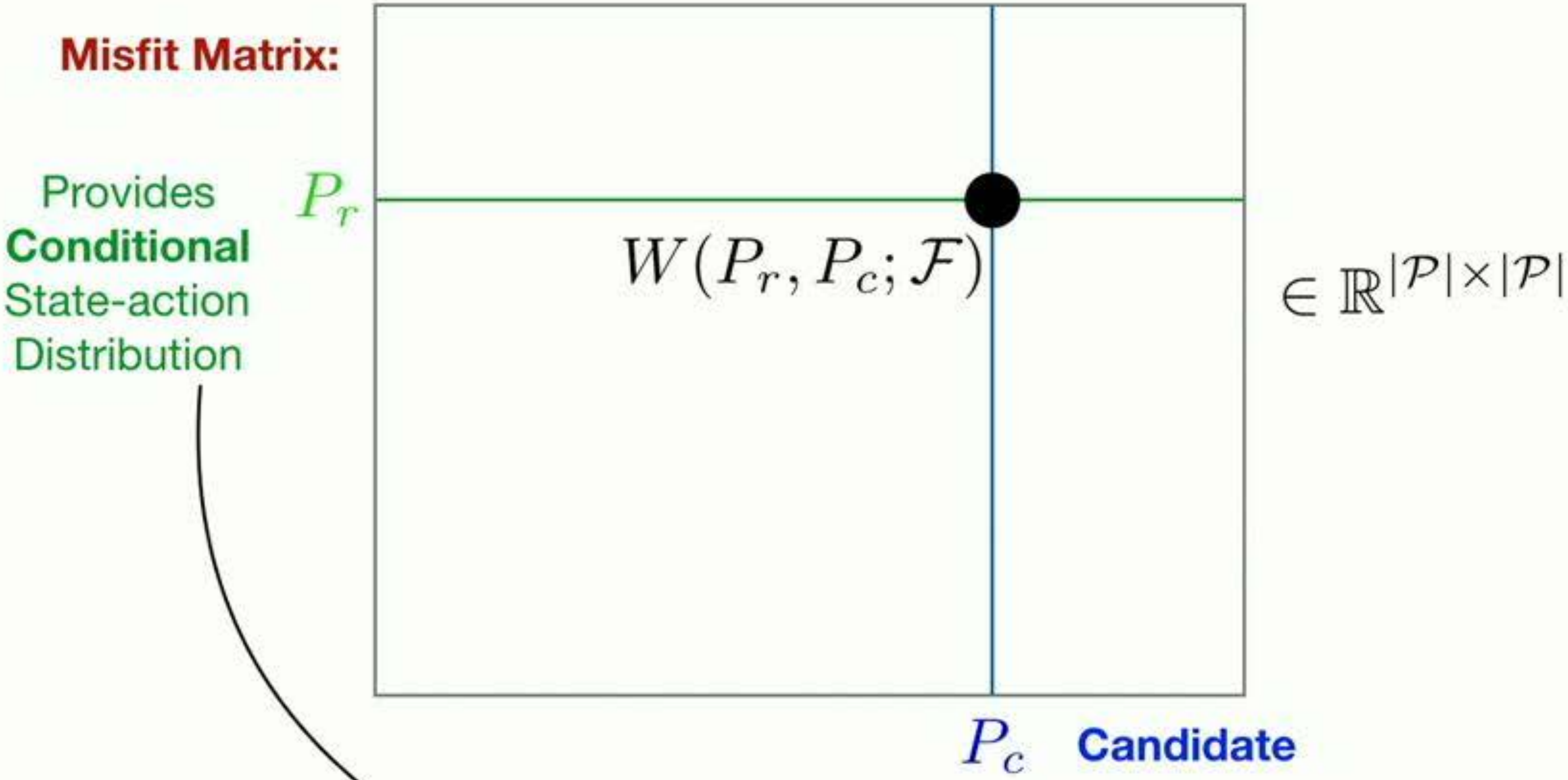


$\in \mathbb{R}^{|\mathcal{P}| \times |\mathcal{P}|}$

$$\mathbb{E}_{x \sim \pi_{P_r}, a \sim U} \left[\mathbb{E}_{x' \sim P_c} f(x, a, x') - \mathbb{E}_{x' \sim P^\star} f(x, a, x') \right]$$

Imaginary — Real-world

Model Rank

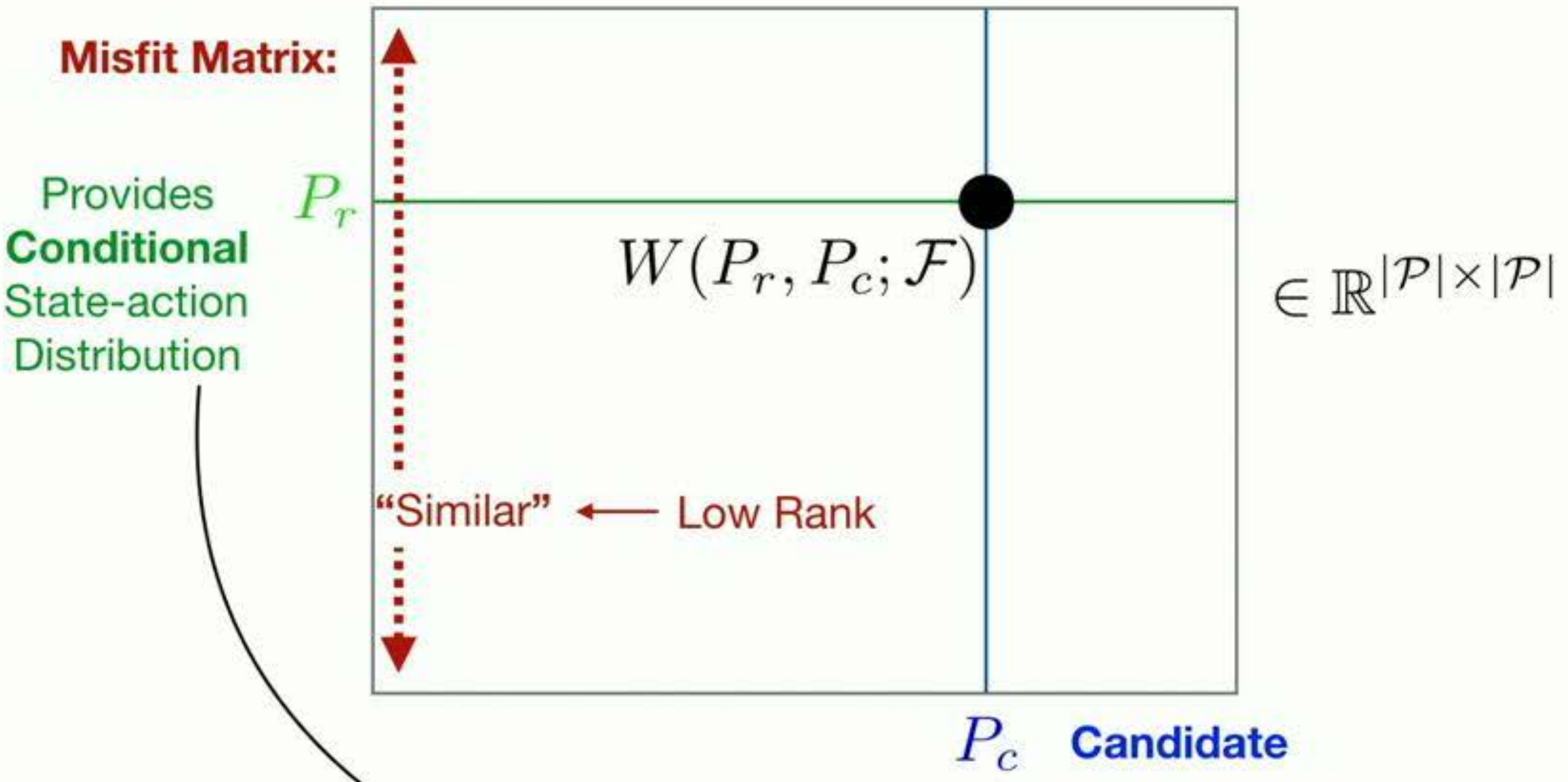


$$W(P_r, P_c; \mathcal{F}) = \max_{f \in \mathcal{F}} \mathbb{E}_{x \sim \pi_{P_r}, a \sim U} \left[\mathbb{E}_{x' \sim P_c} f(x, a, x') - \mathbb{E}_{x' \sim P^\star} f(x, a, x') \right]$$

Imaginary Real-world

Model Rank is defined as the rank of this misfit matrix

Model Rank

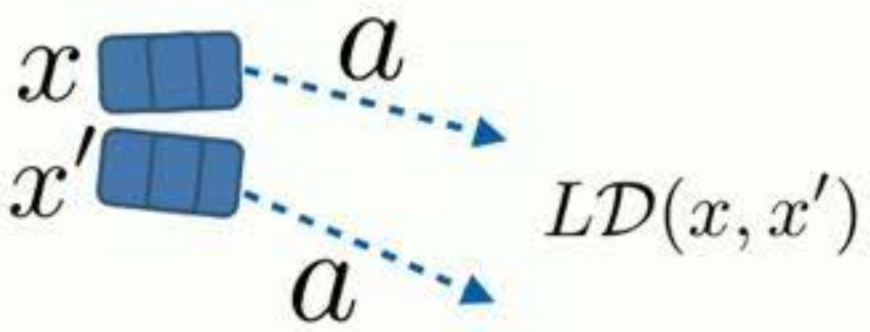


$$W(P_r, P_c; \mathcal{F}) = \max_{f \in \mathcal{F}} \mathbb{E}_{x\sim \pi_{P_r}, a\sim U}\left[\mathbb{E}_{x'\sim P_c} f(x,a,x') - \mathbb{E}_{x'\sim P^\star} f(x,a,x')\right]$$

Imaginary Real-world

Model Rank is defined as the rank of this misfit matrix

A Unified Framework

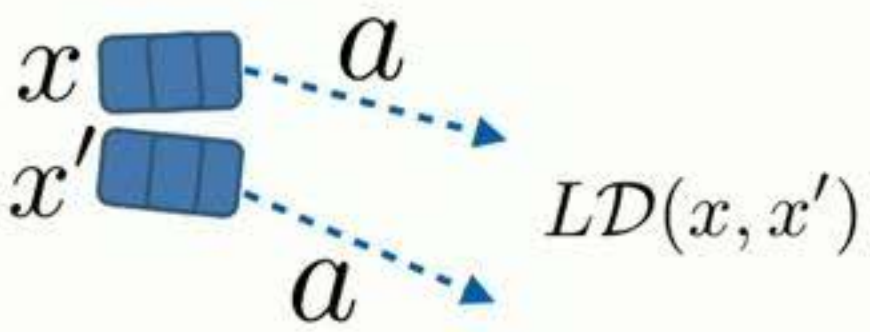


Lipschitz Continuous MDPs

Rank <= Covering number of state space

[KLK, 03]

A Unified Framework



Lipschitz Continuous MDPs

Rank <= Covering number of state space

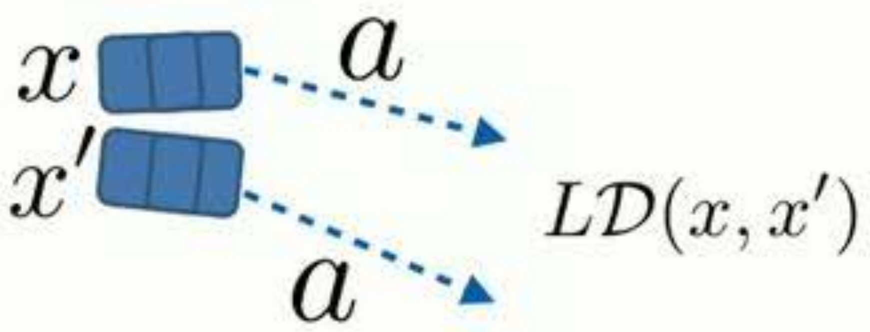
[KLK, 03]

Factored MDPs

Rank <= exp(in-degree)

[GKPV, 03; OV, 13, NIPS]

A Unified Framework



Lipschitz Continuous MDPs

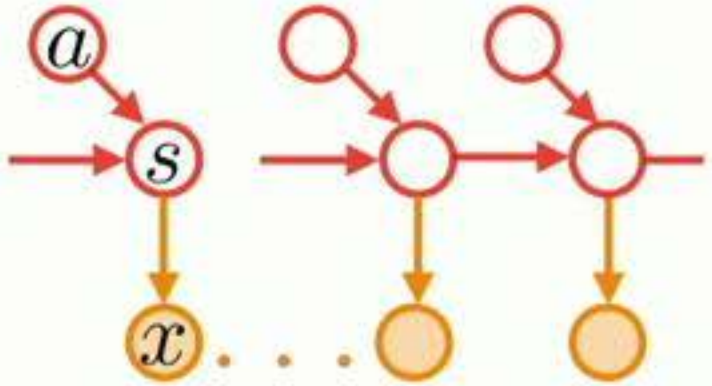
Rank <= Covering number of state space

[KLK, 03]

Factored MDPs

Rank <= exp(in-degree)

[GKPV, 03; OV, 13, NIPS]

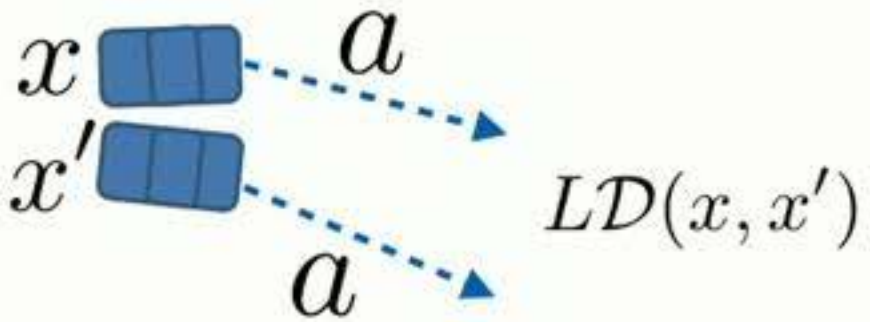


POMDP

Rank <= # of hidden states

[KAL, 16 NIPS]

A Unified Framework



Lipschitz Continuous MDPs

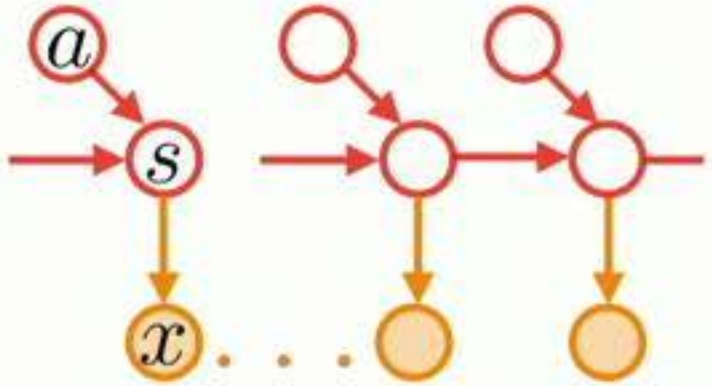
Rank <= Covering number of state space

[KLK, 03]

Factored MDPs

Rank <= exp(in-degree)

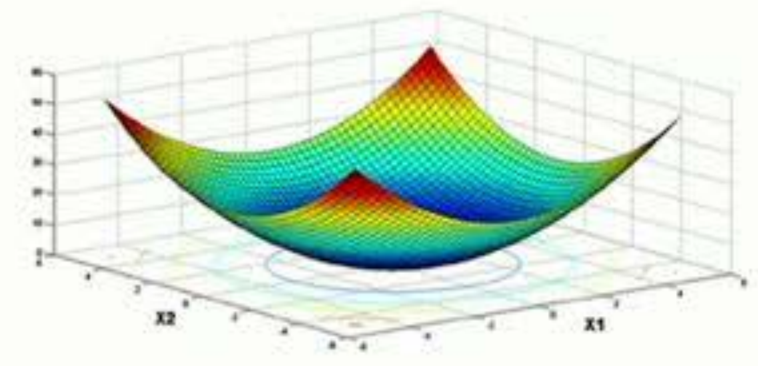
[GKPV, 03; OV, 13, NIPS]



POMDP

Rank <= # of hidden states

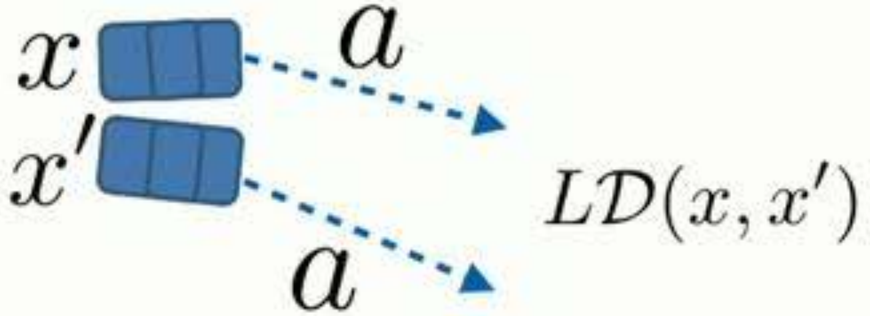
[KAL, 16 NIPS]



Linear Quadratic Regulator

Rank = O(d^2)

A Unified Framework



Lipschitz Continuous MDPs

Rank <= Covering number of state space

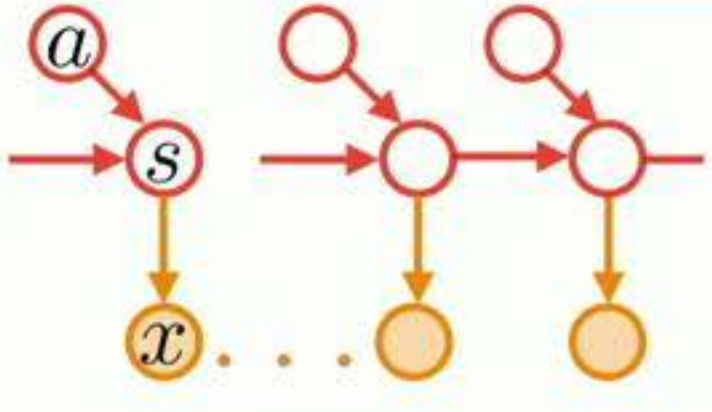
[KLK, 03]

Factored MDPs

<= exp(in-degree)

KPV, 03; OV, 13, NIPS]

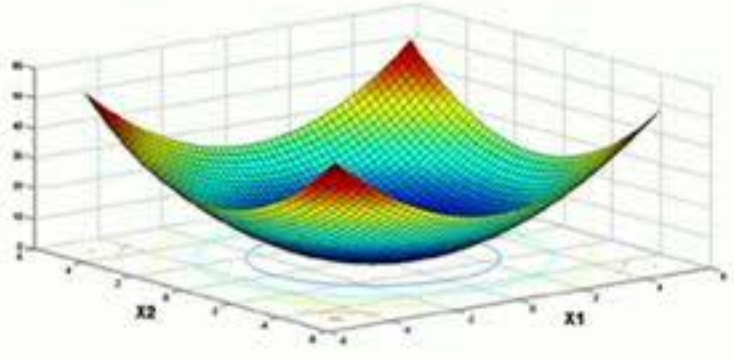
A Unified Algorithm!



POMDP

Rank <= # of hidden states

[KAL, 16 NIPS]



Linear Quadratic Regulator

Rank = O(d^2)

Sample Complexity

$$\tilde{O}\Big(\frac{H^3 R^2 |\mathcal{A}|}{\epsilon^2} \log\Big(\frac{|\mathcal{F}||\mathcal{P}|}{\delta}\Big)\Big)$$

Model-based Reinforcement Learning in Contextual Decision Processes
[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Sample Complexity

Model Rank

$$\tilde{O}\left(\frac{H^3 R^2 |\mathcal{A}|}{\epsilon^2} \log\left(\frac{|\mathcal{F}||\mathcal{P}|}{\delta}\right)\right)$$

Complexity of Discriminators & Models

~~Poly Dependency on # of States~~

Model-based Reinforcement Learning in Contextual Decision Processes
[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Sample Complexity

Model Rank

$$\tilde{O}\Big(\frac{H^3 R^2 |\mathcal{A}|}{\epsilon^2} \log\Big(\frac{|\mathcal{F}||\mathcal{P}|}{\delta}\Big)\Big)$$

Complexity of Discriminators & Models

~~Poly Dependency on # of States~~

Supervised Learning Type Generalization !

Model-based Reinforcement Learning in Contextual Decision Processes
[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Generalization & Sample Efficiency via…

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]

[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]

All RL Problems

Sample Complexity

$$\tilde{O}\left(\frac{H^3 R^2 |\mathcal{A}|}{\epsilon^2} \log\left(\frac{|\mathcal{F}||\mathcal{P}|}{\delta}\right)\right)$$

Model-based Reinforcement Learning in Contextual Decision Processes
[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Sample Complexity

Model Rank

$$\tilde{O}\Big(\frac{H^3 R^2 |\mathcal{A}|}{\epsilon^2} \log\Big(\frac{|\mathcal{F}||\mathcal{P}|}{\delta}\Big)\Big)$$

Complexity of Discriminators & Models

~~Poly Dependency on # of States~~

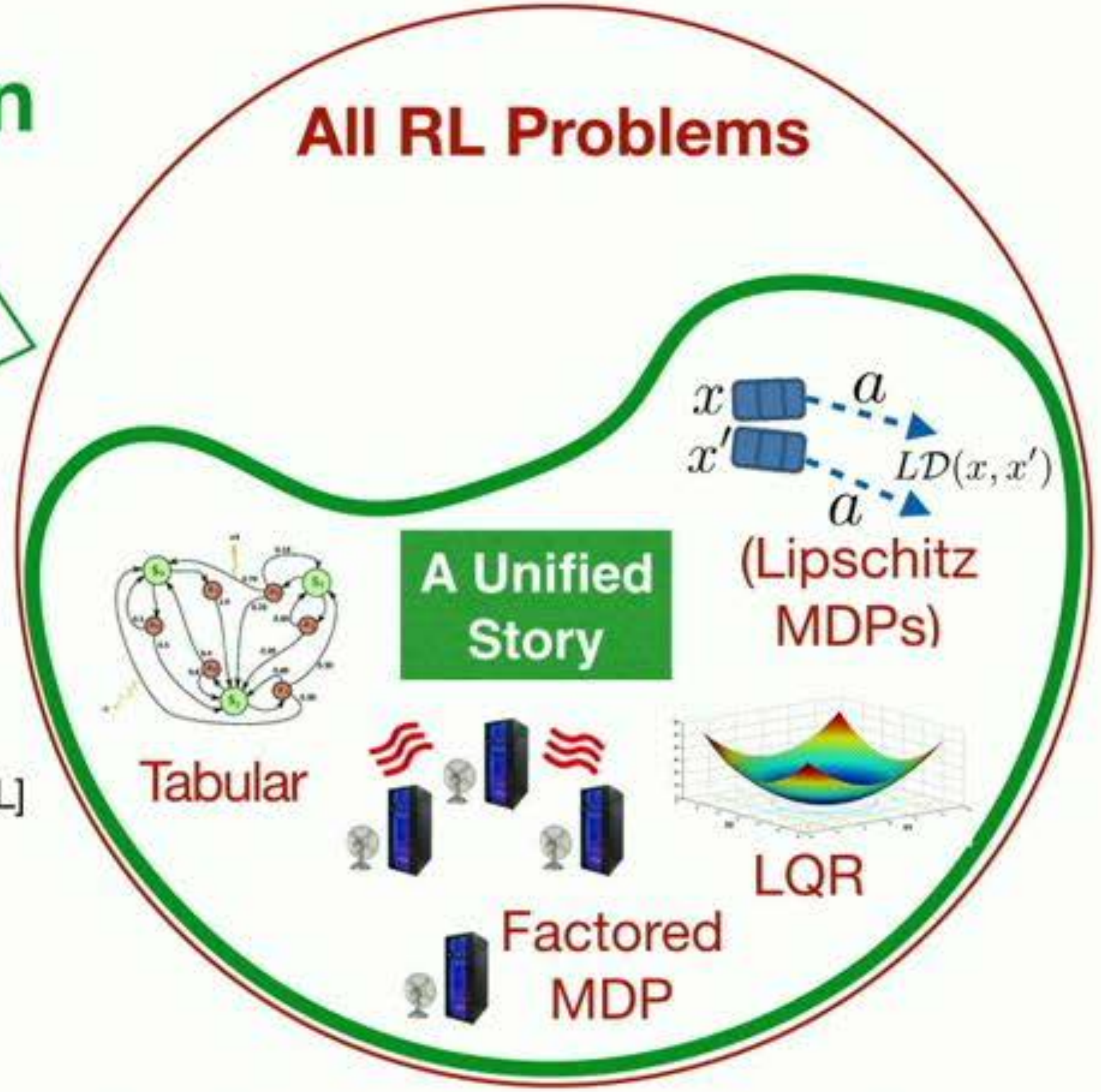
Model-based Reinforcement Learning in Contextual Decision Processes
[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Generalization & Sample Efficiency via...

1. Expert Demonstration

[**Sun**, Venkatraman, Gordon, Boots, Bagnell, 17, ICML]

[**Sun**, Gordon, Boots, Bagnell, 18, NeurIPS]



2. Exploiting Structures

[**Sun**, Jiang, Krishnamurthy, Agarwal, Langford, arXiv, 18]

Future Work

Medical Treatment

Education

Autonomous Driving

Waseda University's Manga club

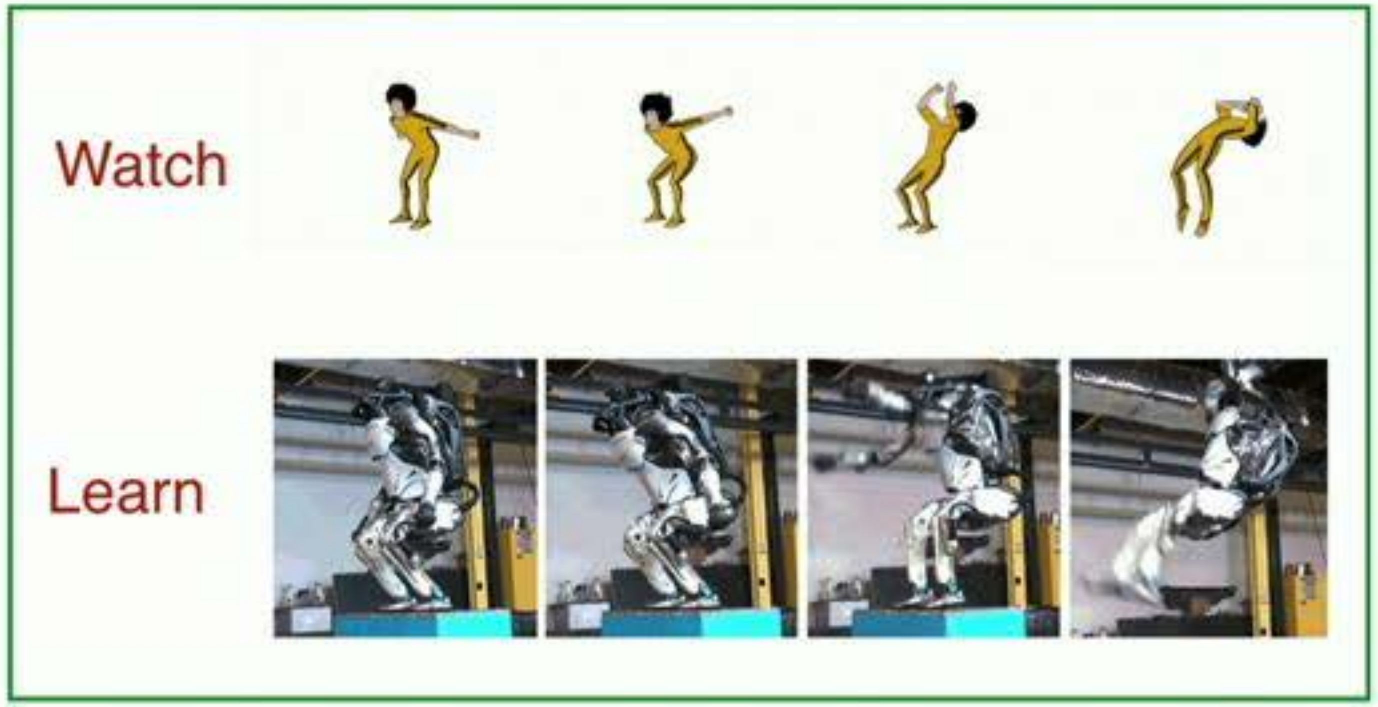
Assistance in Disaster Recovery

1. Leverage Expert Demonstrations

1. Leverage Expert Demonstrations

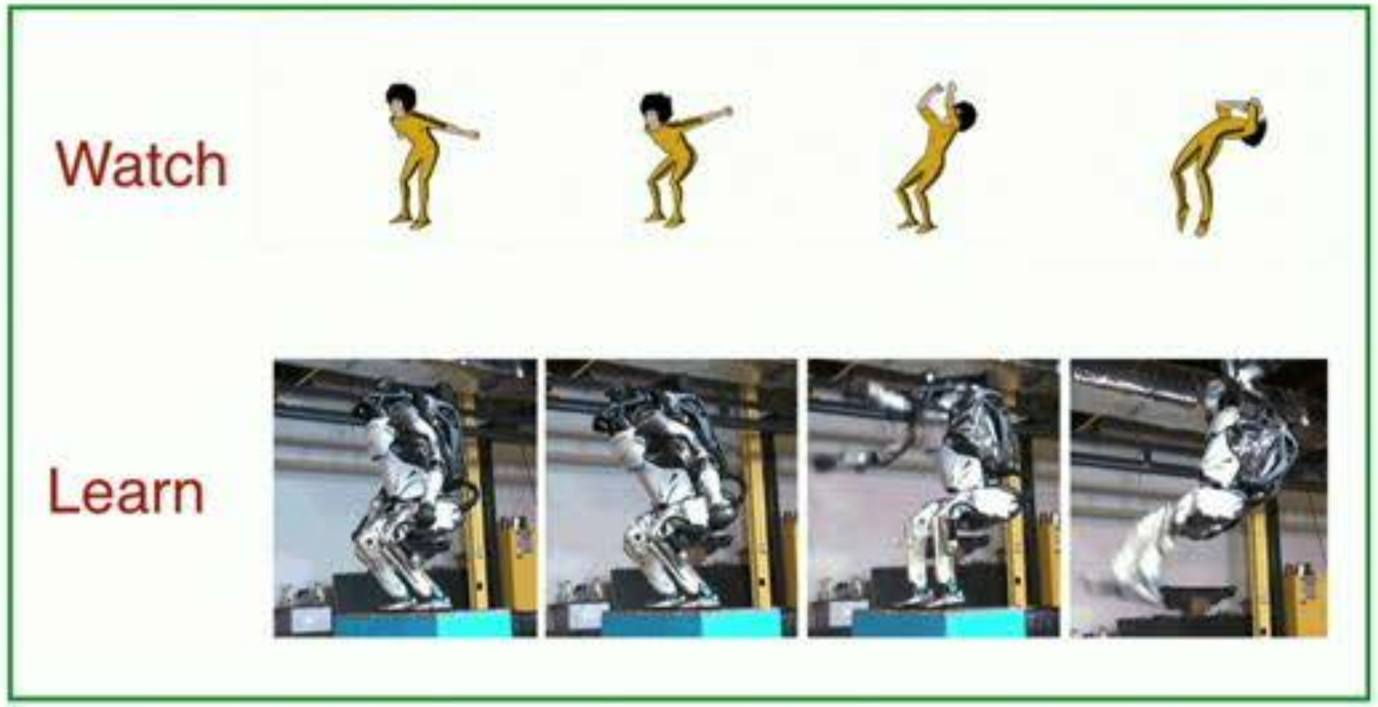
Interactive Imitation Learning

1. Leverage Expert Demonstrations



1. Leverage Expert Demonstrations

No Interaction
No Expert Action
No Reward

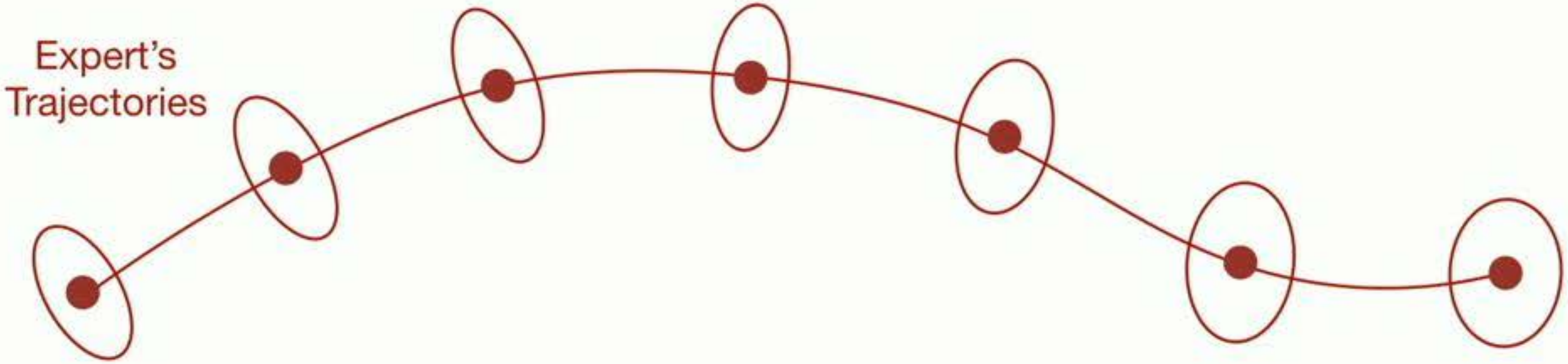


Imitation Learning from Observations

Forward Adversarial Imitation Learning (FAIL):

[**Sun** et.al, In Submission, 19]

Learn policies using Integral Probability Metric

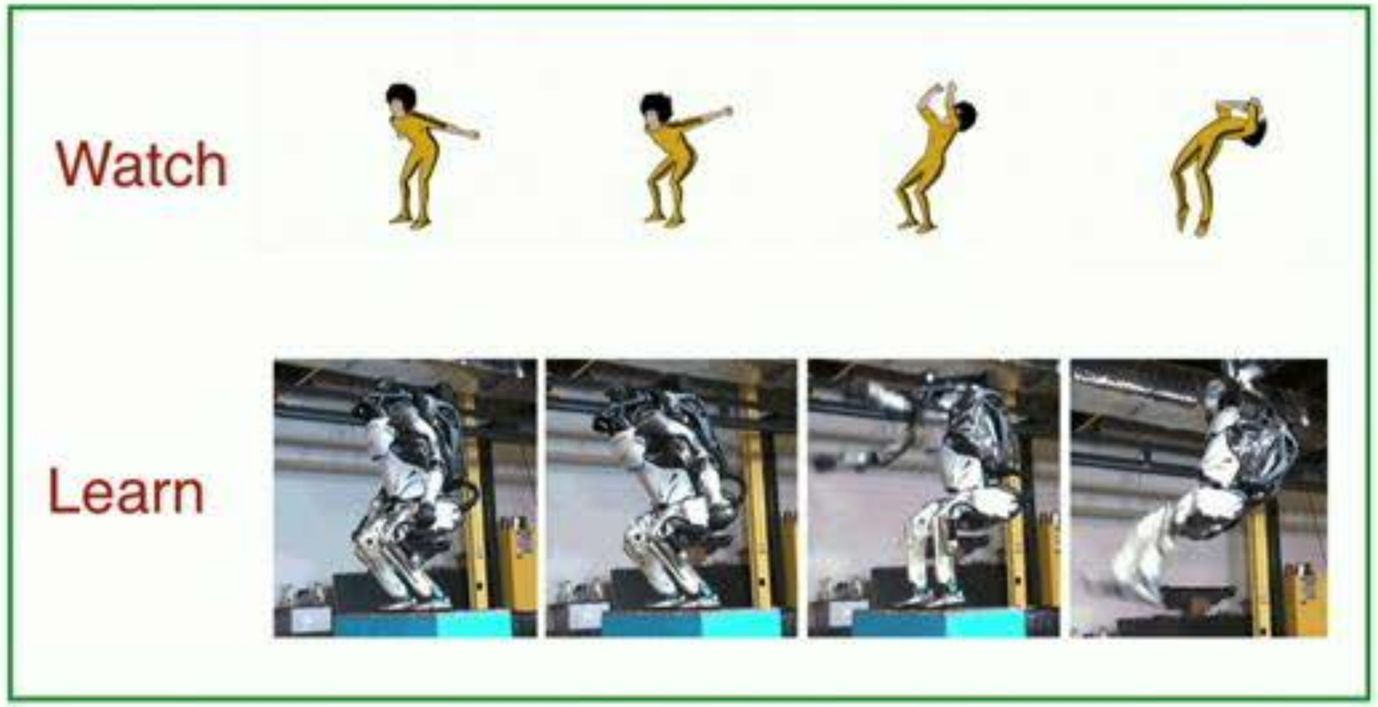


1. Leverage Expert Demonstrations

No Interaction

No Expert Action

No Reward

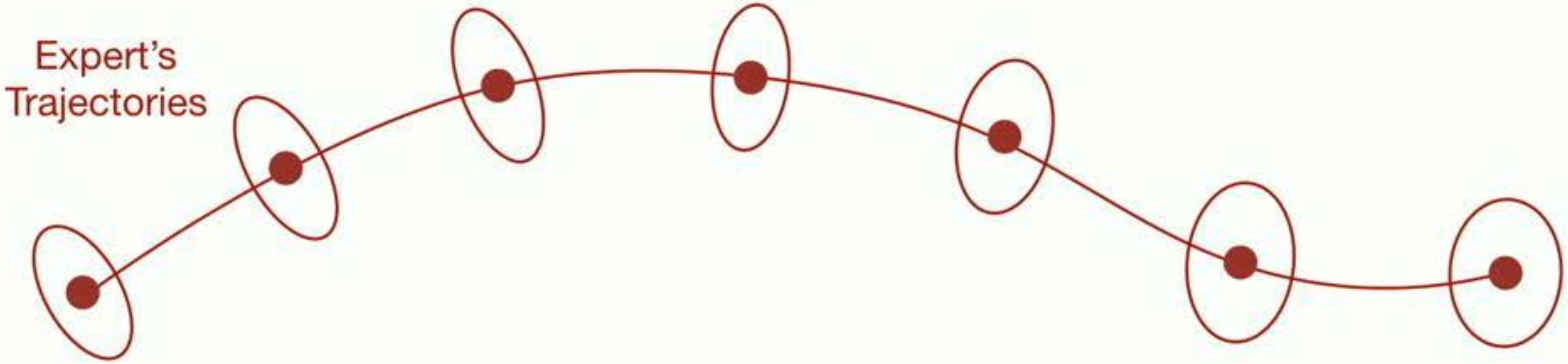


Imitation Learning from Observations

Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

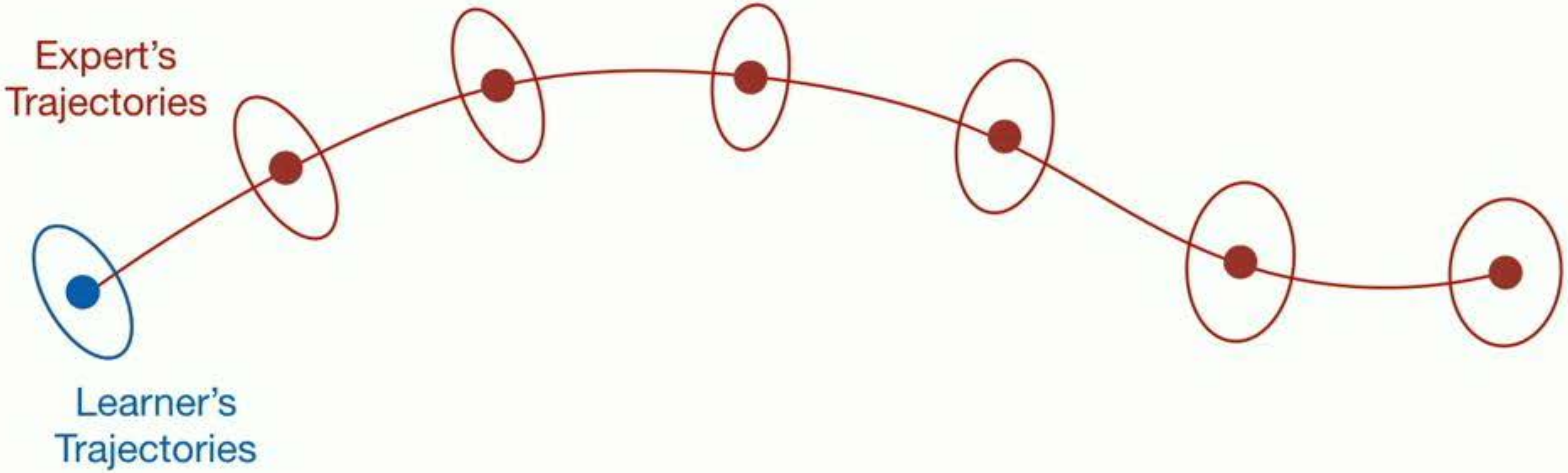
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

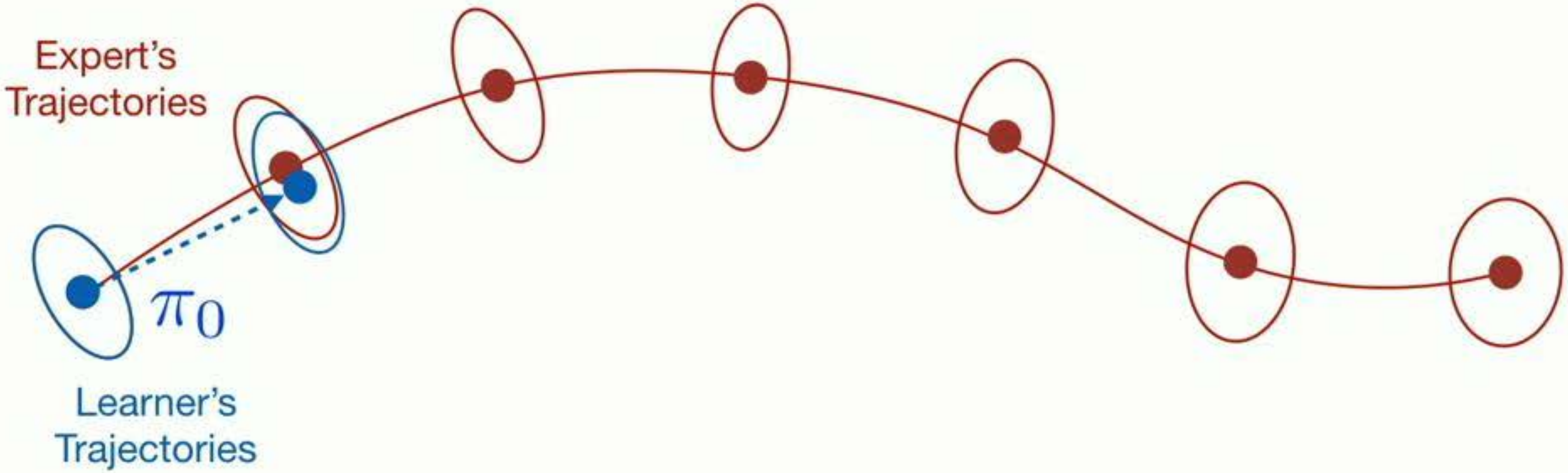
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

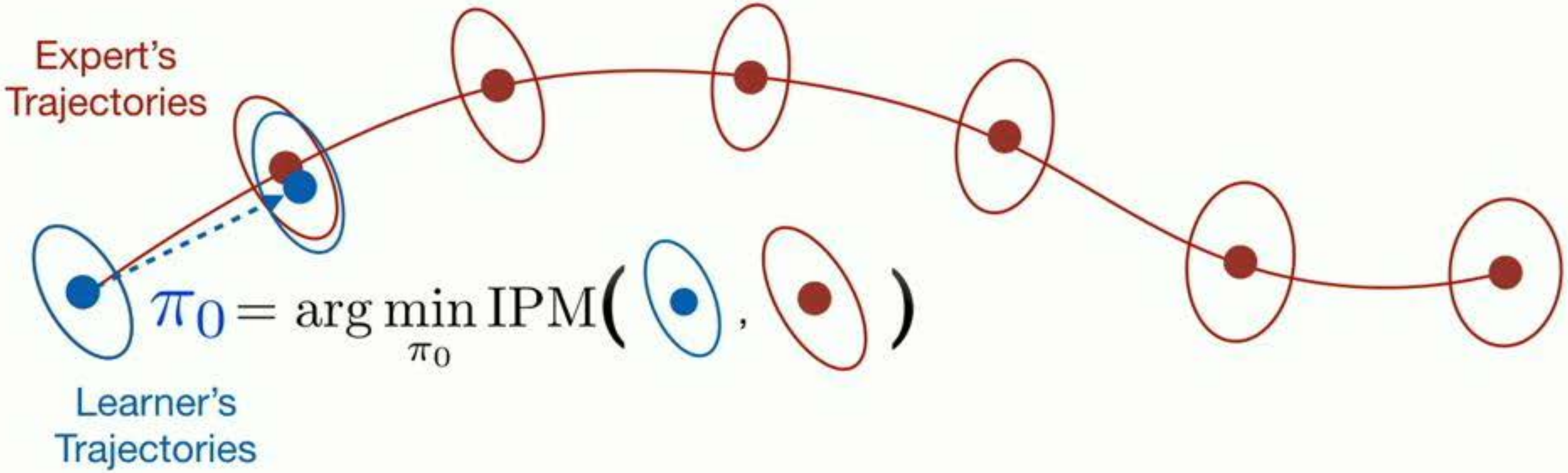
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[**Sun** et.al, In Submission, 19]

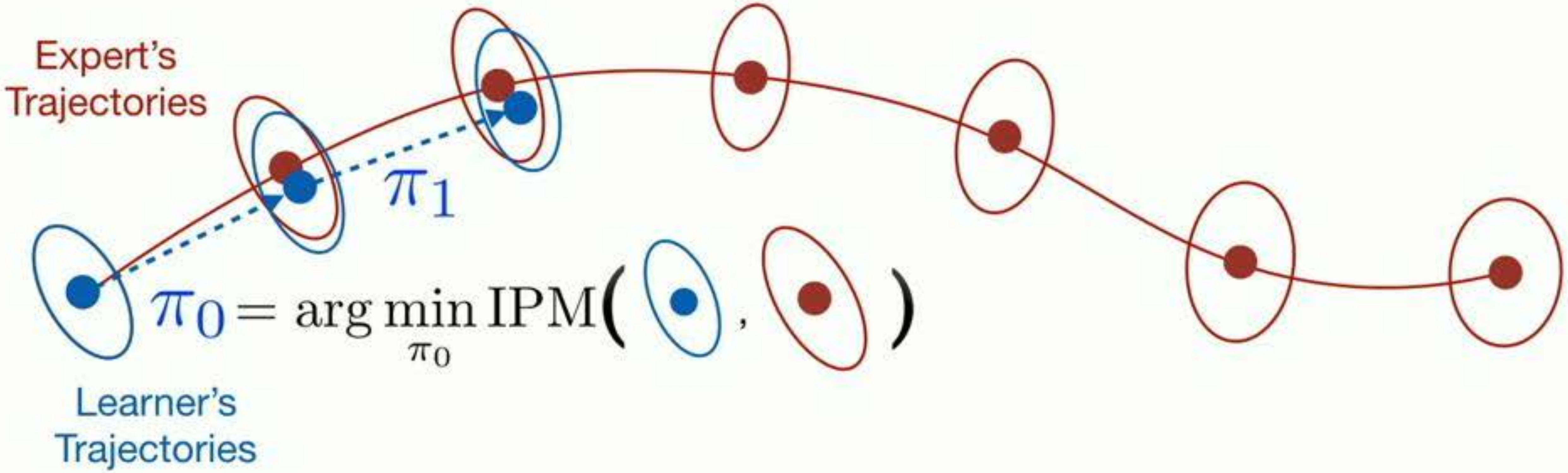
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[**Sun** et.al, In Submission, 19]

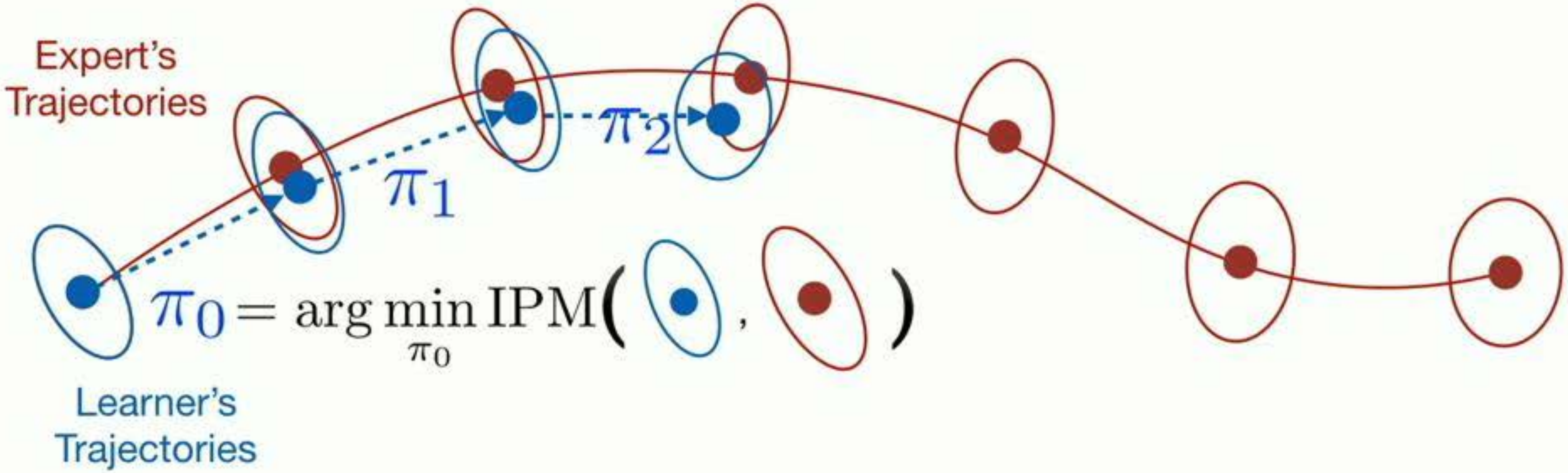
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

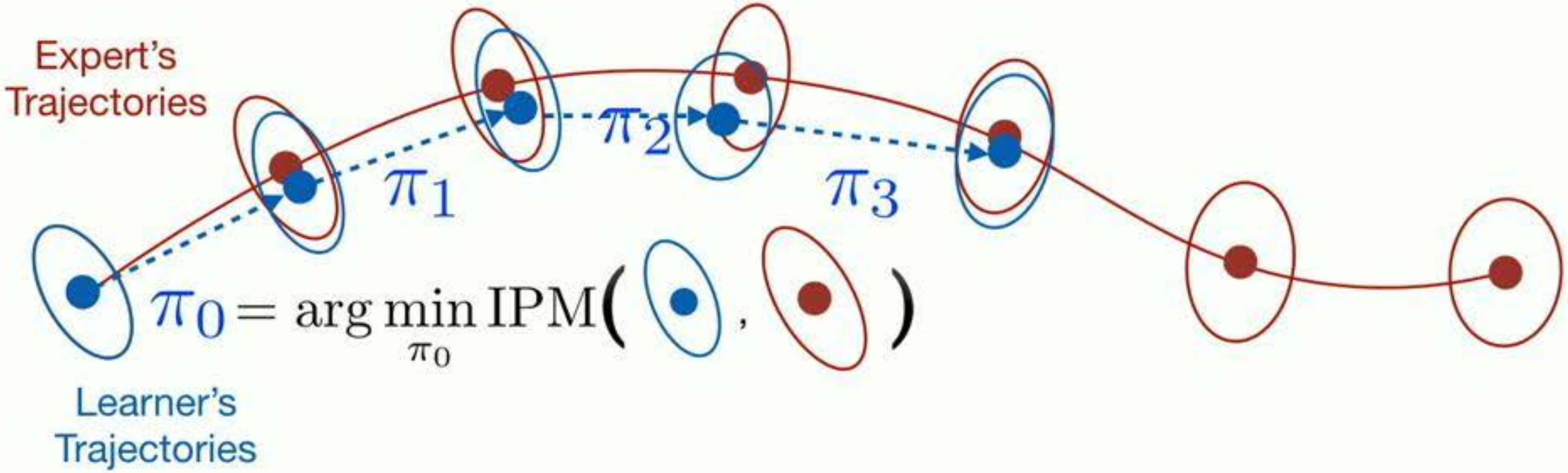
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

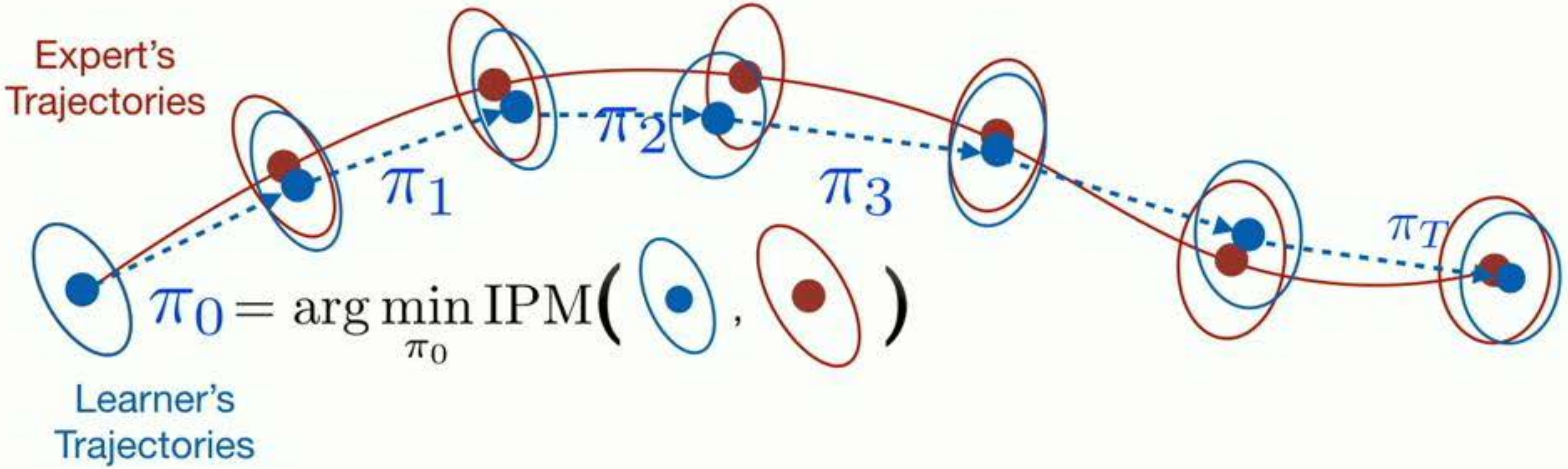
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

[Sun et.al, In Submission, 19]

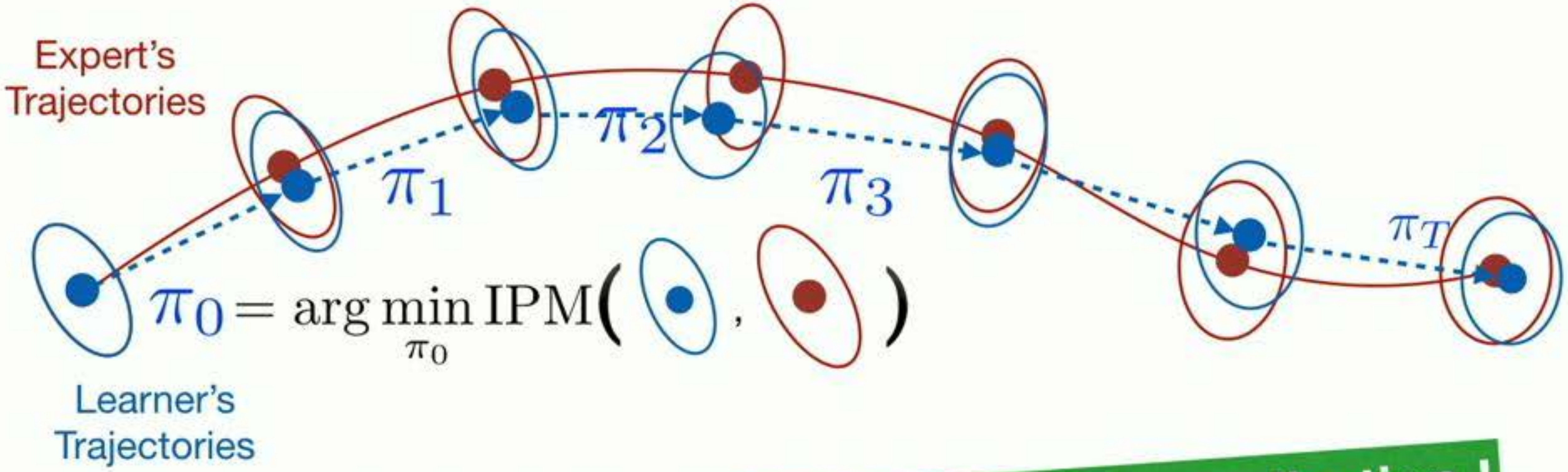
Learn policies using Integral Probability Metric



Forward Adversarial Imitation Learning (FAIL):

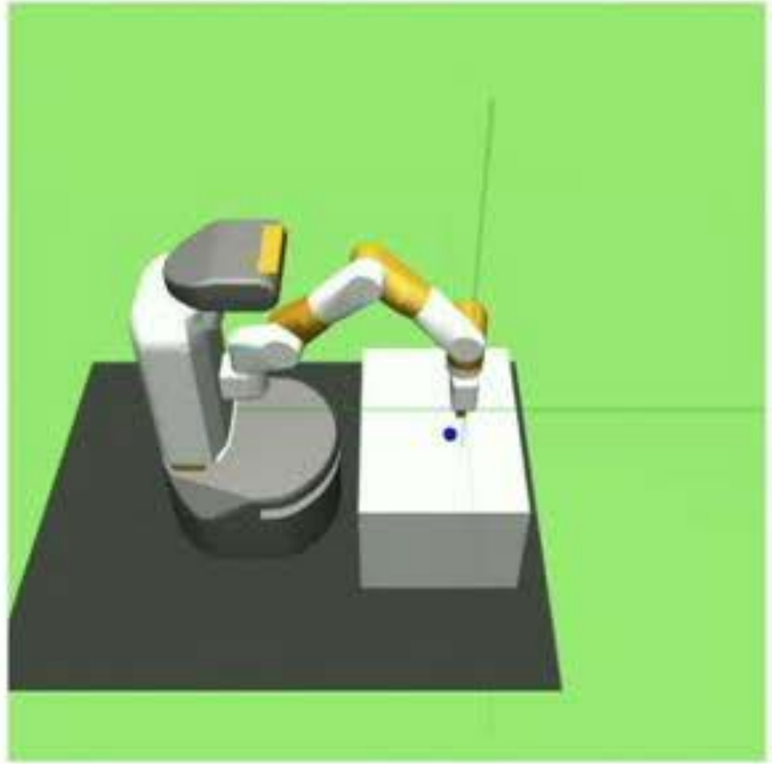
[Sun et.al, In Submission, 19]

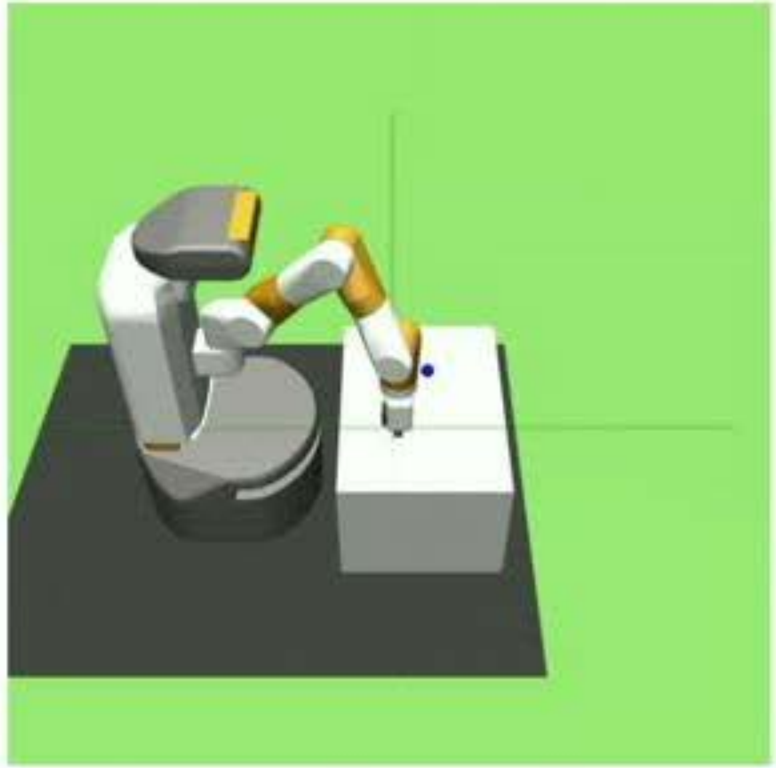
Learn policies using Integral Probability Metric



Supervised Learning Type Generalization !

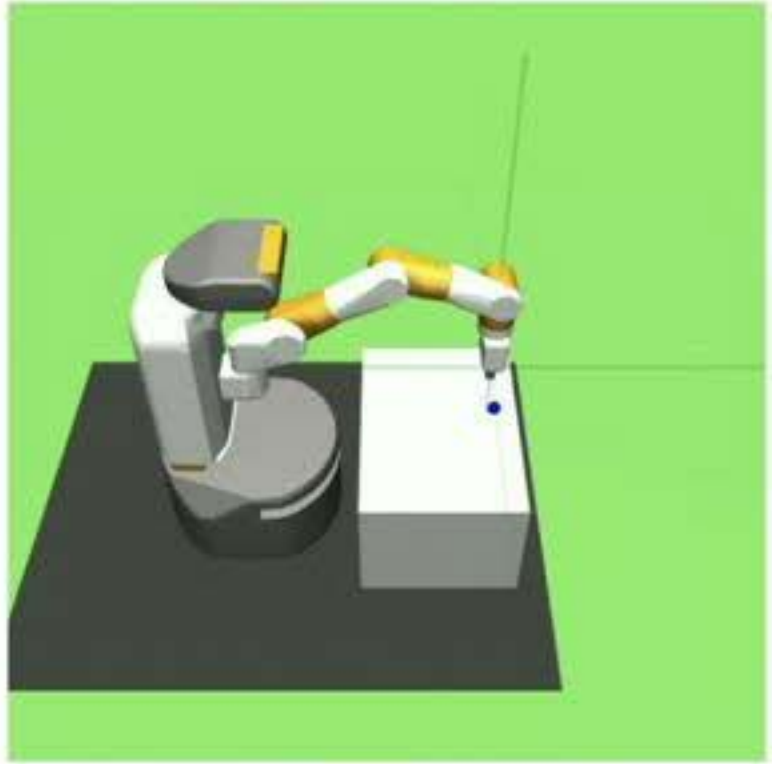
Promising Simulation Results…

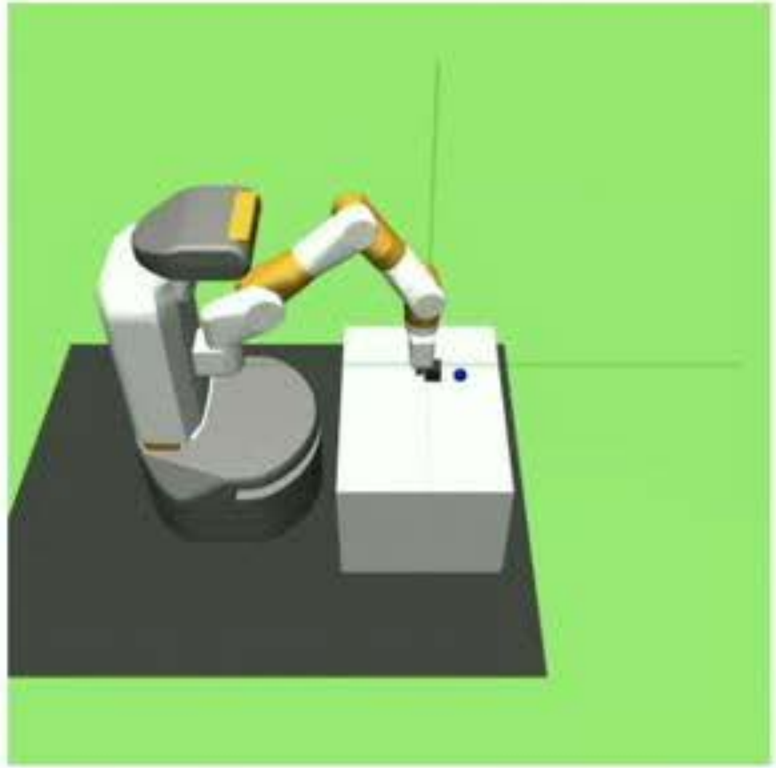




[Fetch Robot Simulator from OpenAI Gym]

Promising Simulation Results…

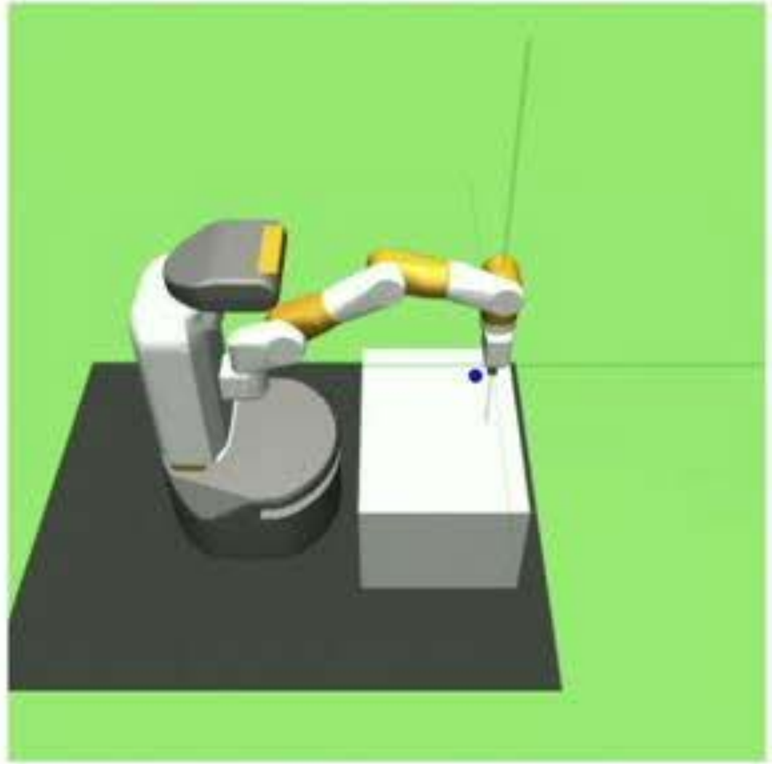


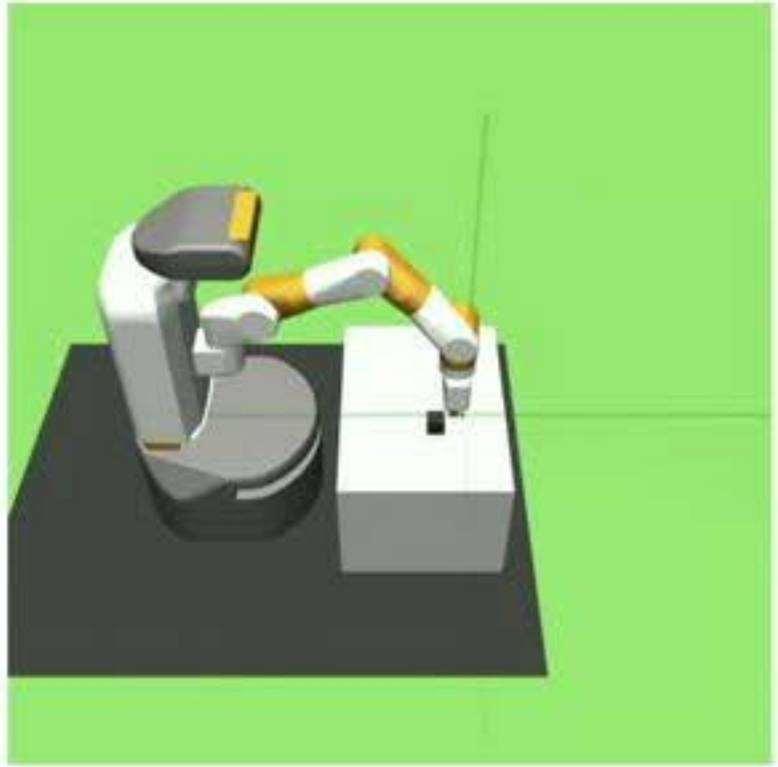


[Fetch Robot Simulator from OpenAI Gym]

Image from https://www.asme.org/engineering-topics/articles/robotics/robots-kitchen-at-the-table

Promising Simulation Results…





[Fetch Robot Simulator from OpenAI Gym]

Image from https://www.asme.org/engineering-topics/articles/robotics/robots-kitchen-at-the-table

Lots of Challenges:

- Learn from videos
- Interaction with experts

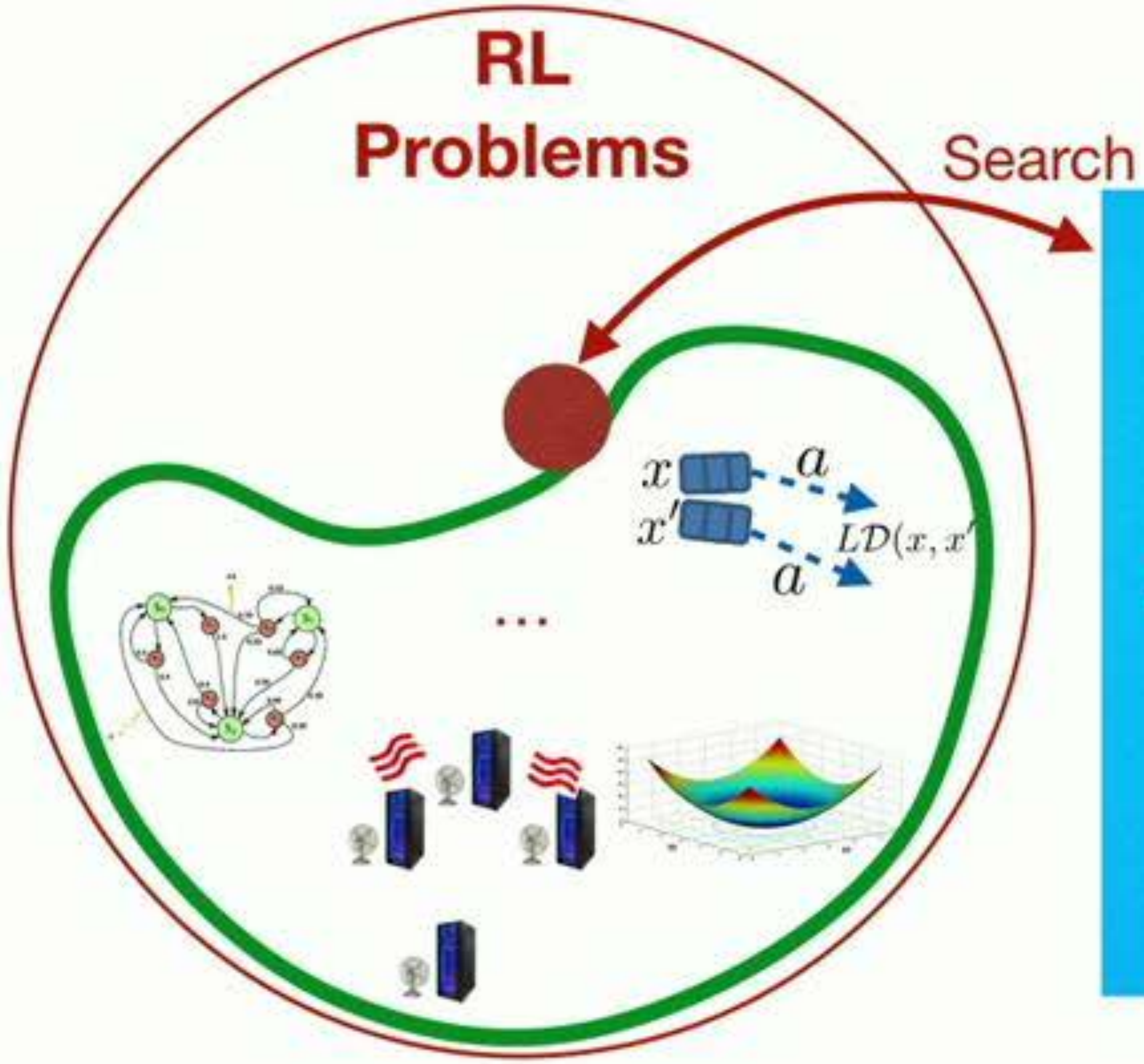
2. Generalization from Prior Experiences

Medical Treatment

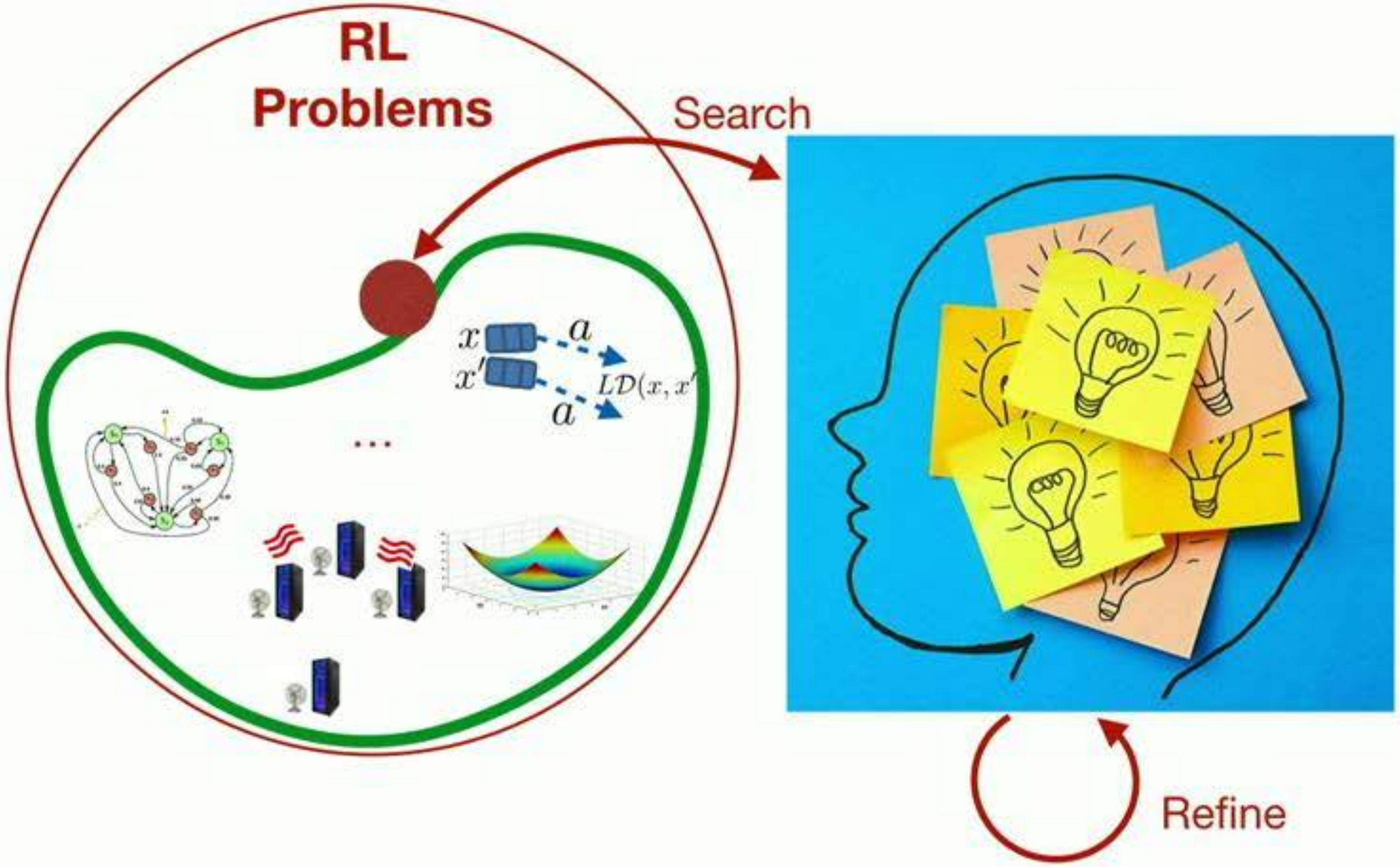
Waseda University's Manga club

Assistance in Disaster Recovery

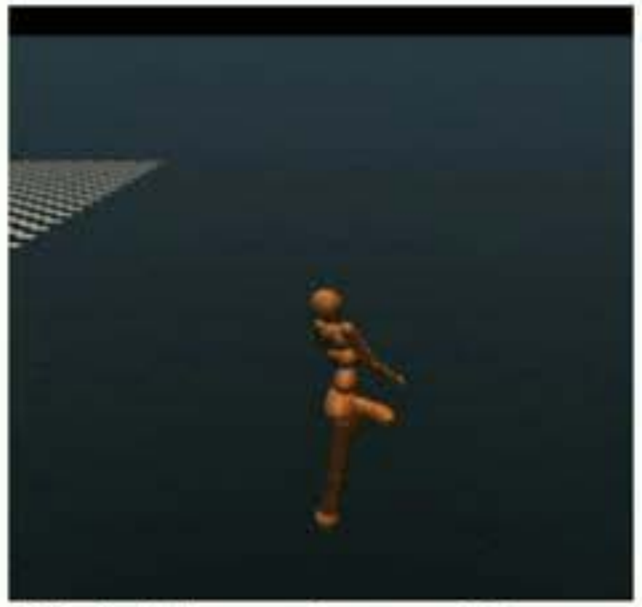
2. Generalization from Prior Experiences



2. Generalization from Prior Experiences



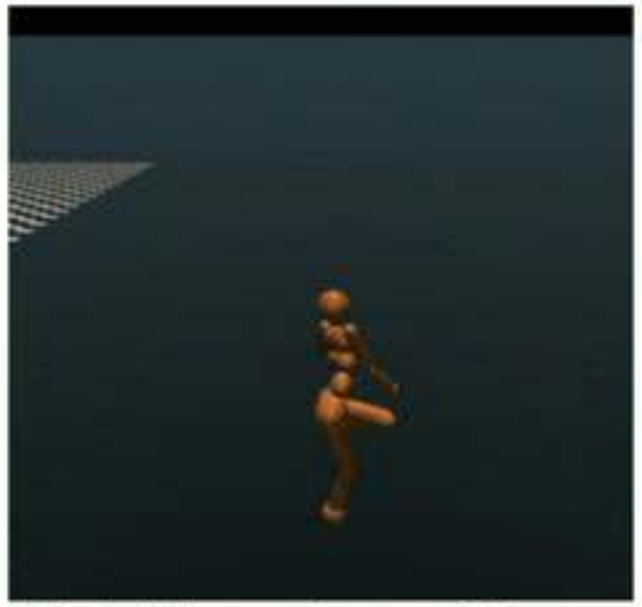
2. Generalization from Prior Experiences



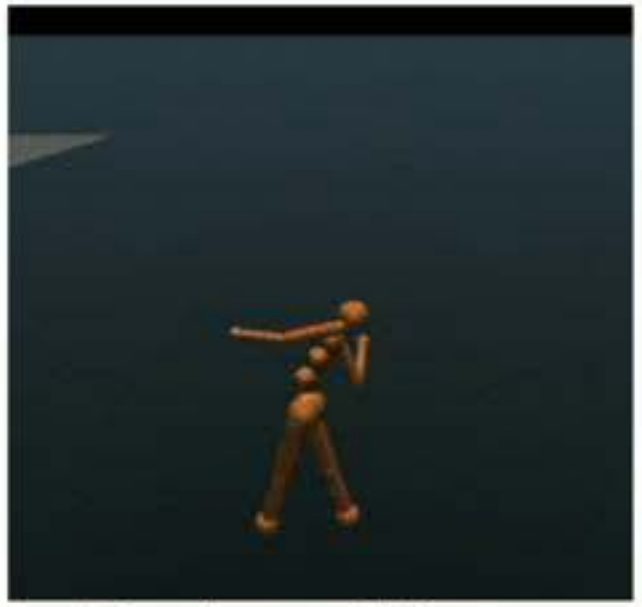
Right Leg Jump Demo

Videos from Ben Recht's Blog (http://www.argmin.net/2018/03/20/mujocoloco/)

2. Generalization from Prior Experiences



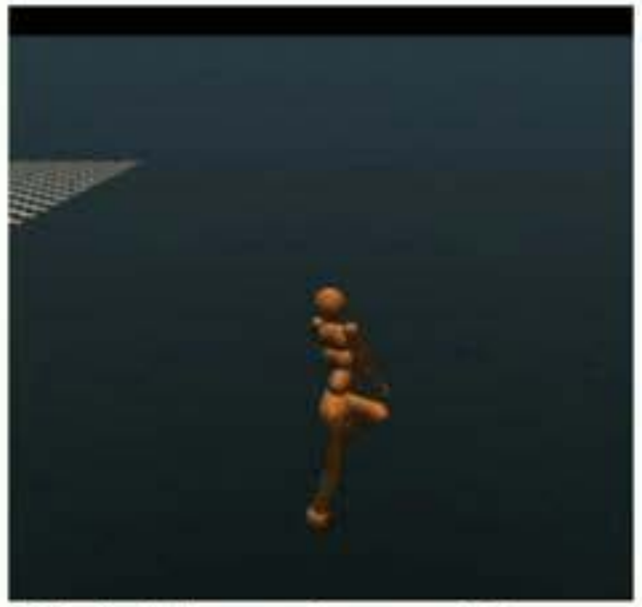
Right Leg Jump Demo



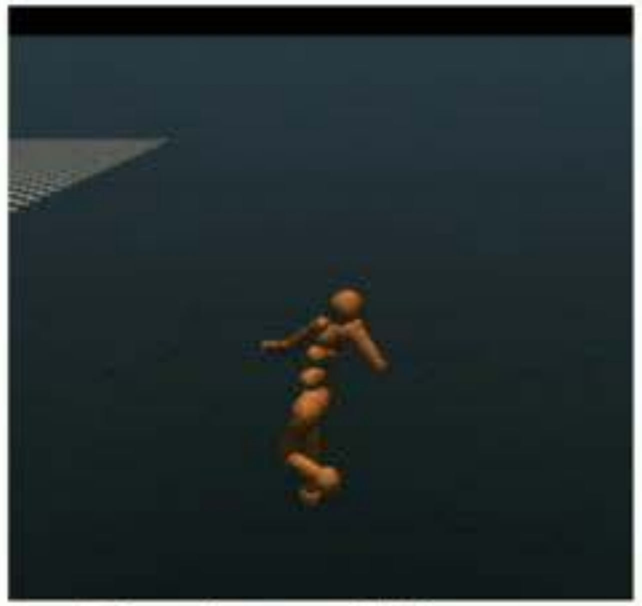
Backward Demo

Videos from Ben Recht's Blog (http://www.argmin.net/2018/03/20/mujocoloco/)

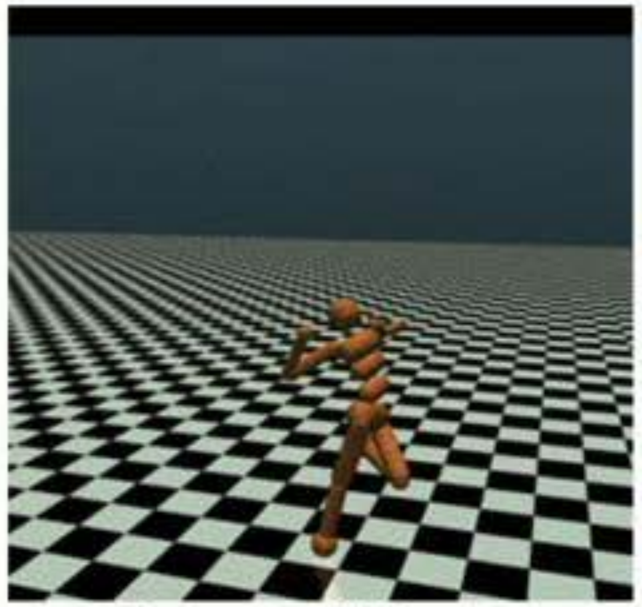
2. Generalization from Prior Experiences



Right Leg Jump Demo



Backward Demo



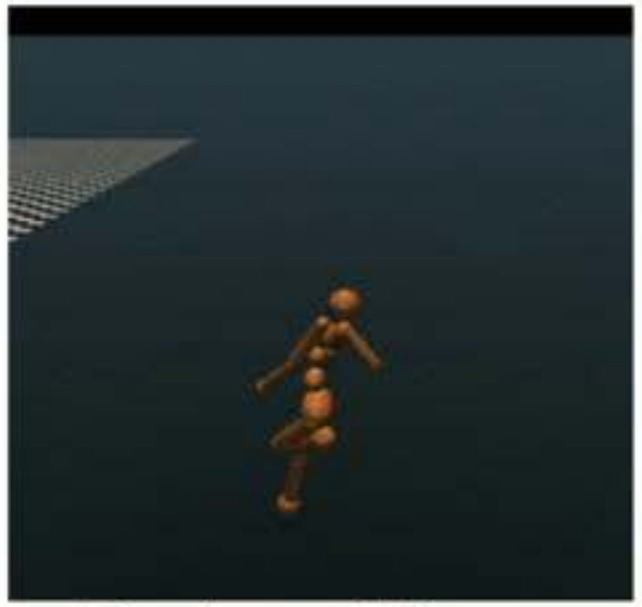
Forward Demo

...

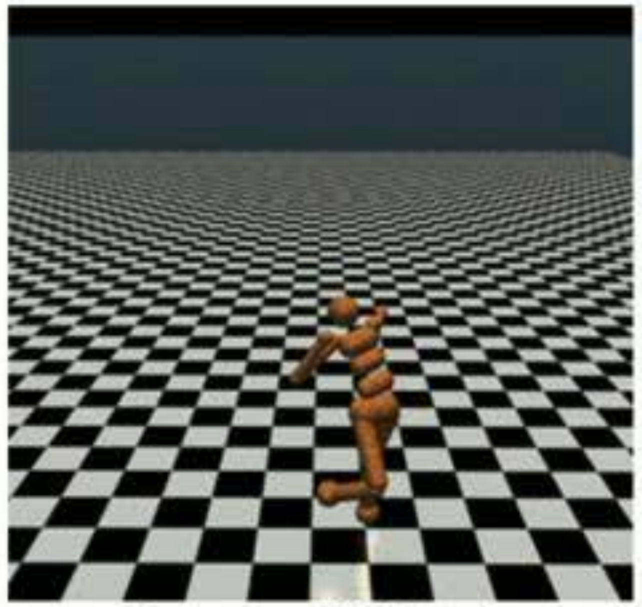
Videos from Ben Recht's Blog (http://www.argmin.net/2018/03/20/mujocoloco/)

2. Generalization from Prior Experiences

Right Leg Jump Demo



Backward Demo

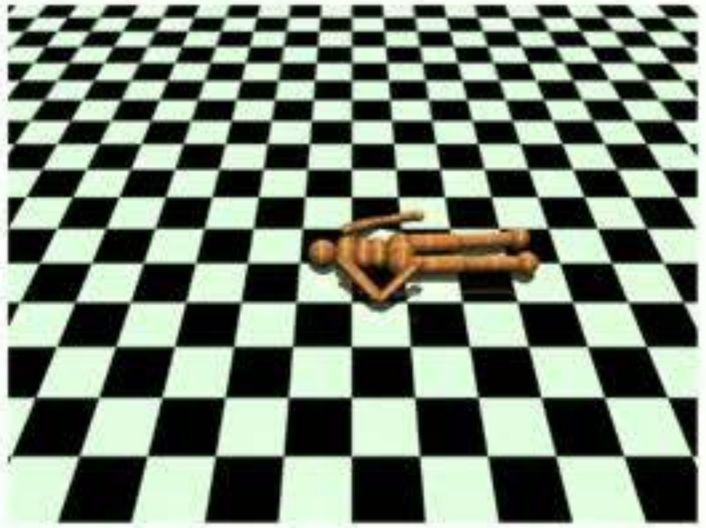


Forward Demo

...

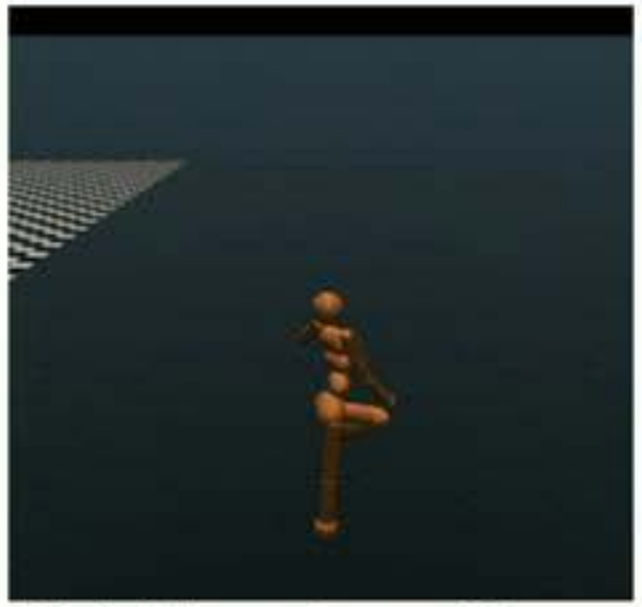
New task:

Stand up with little to no training?



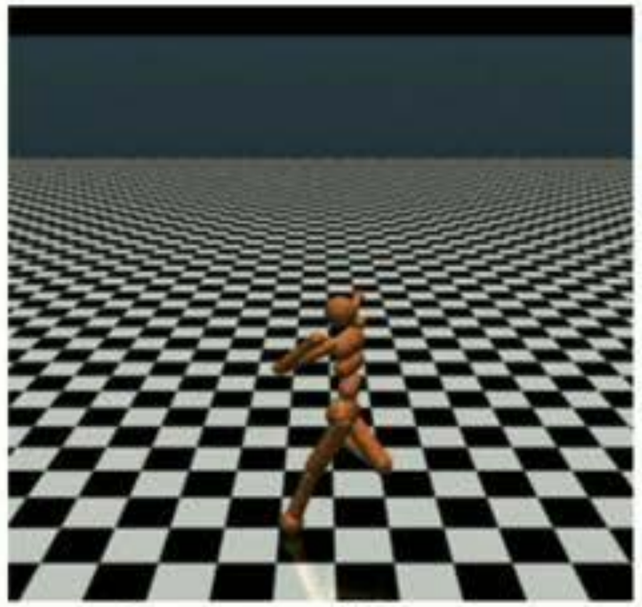
Videos from Ben Recht's Blog (http://www.argmin.net/2018/03/20/mujocoloco/)

2. Generalization from Prior Experiences



Right Leg Jump Demo

Backward Demo

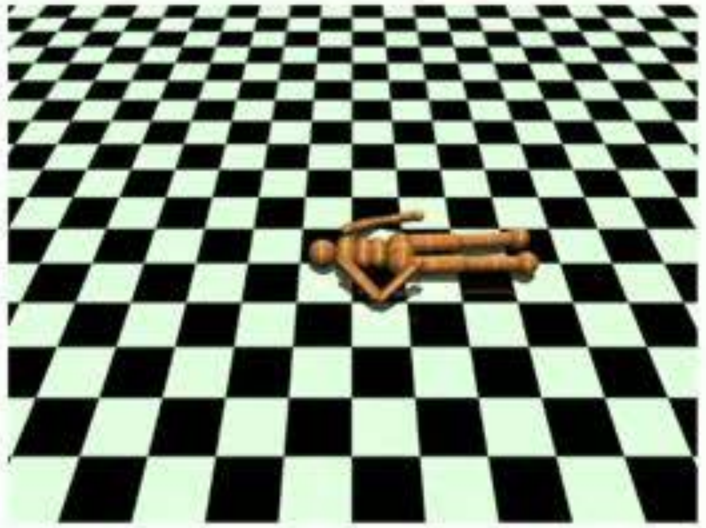


Forward Demo

...

Offline Learning From Prior Relevant Experiences

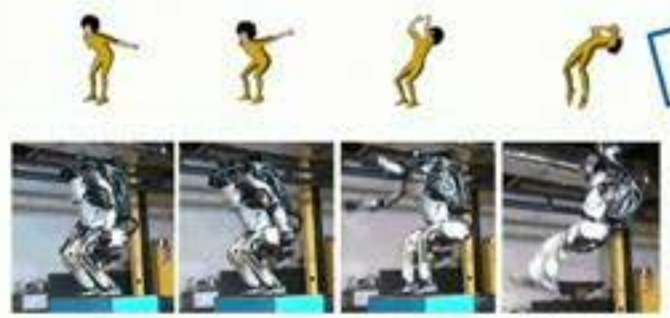
New task:
Stand up with little to no training?

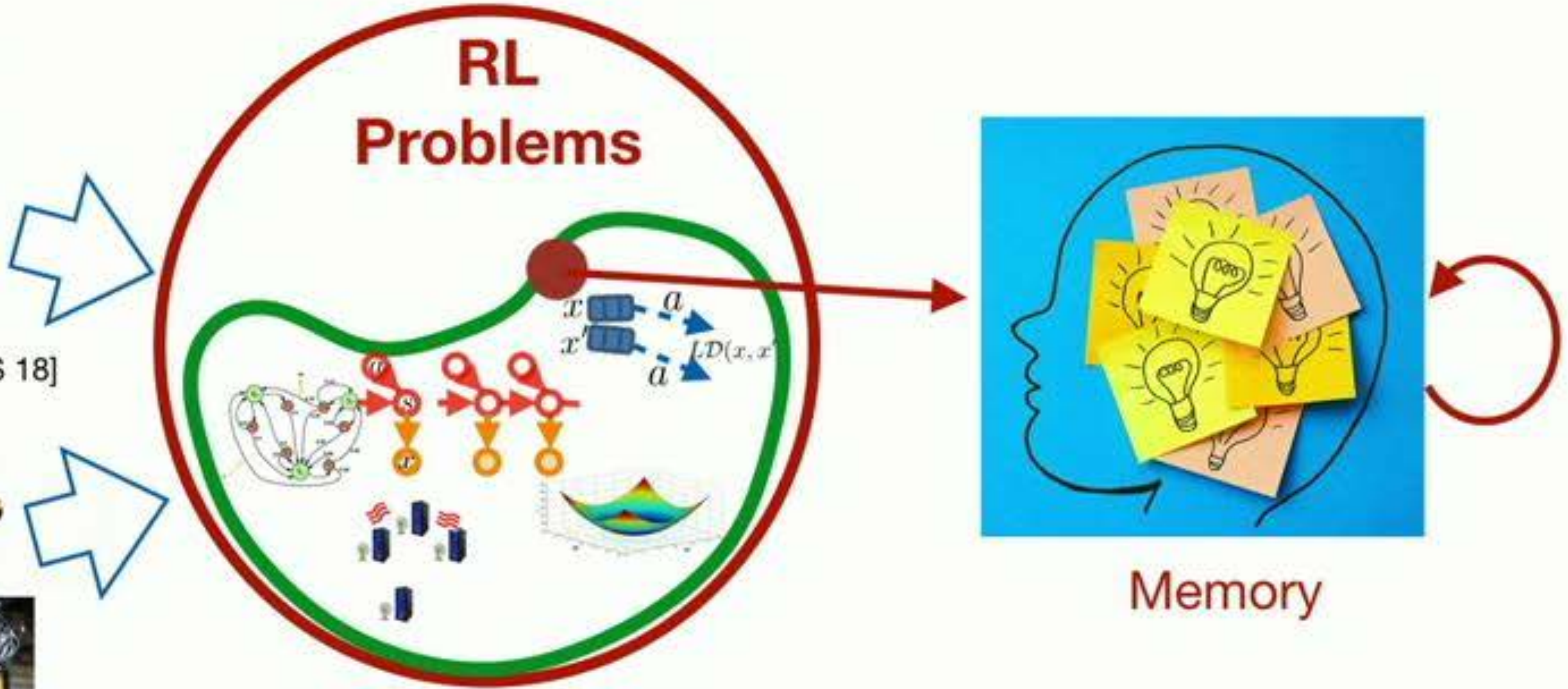


Videos from Ben Recht's Blog (http://www.argmin.net/2018/03/20/mujocoloco/)

Imitation

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





[Sun et.al., 18, arXiv]

Memory

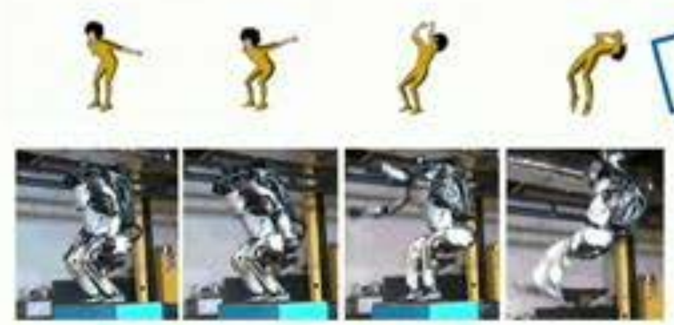
Online Policy Evaluation

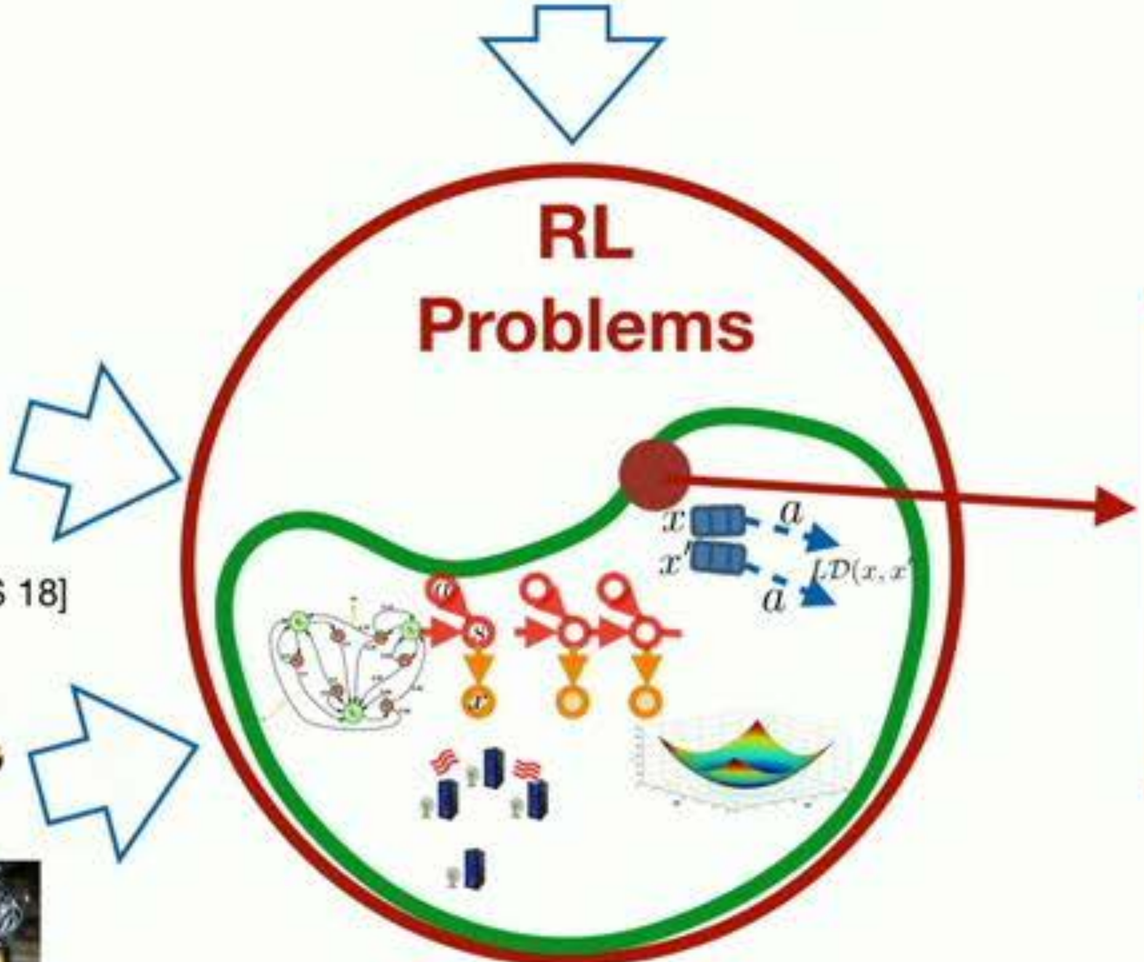
Reduction from Policy Evaluation to No-Regret Online Learning

[Sun, Bagnell, UAI 15, Best Student Paper]

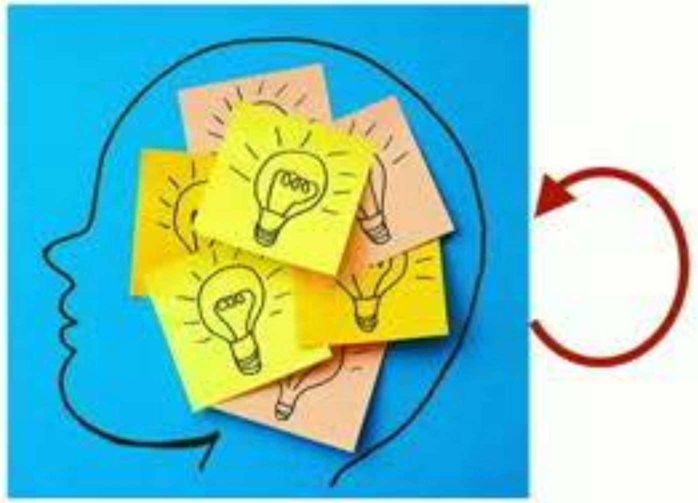
Imitation

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





[Sun et.al., 18, arXiv]



Memory

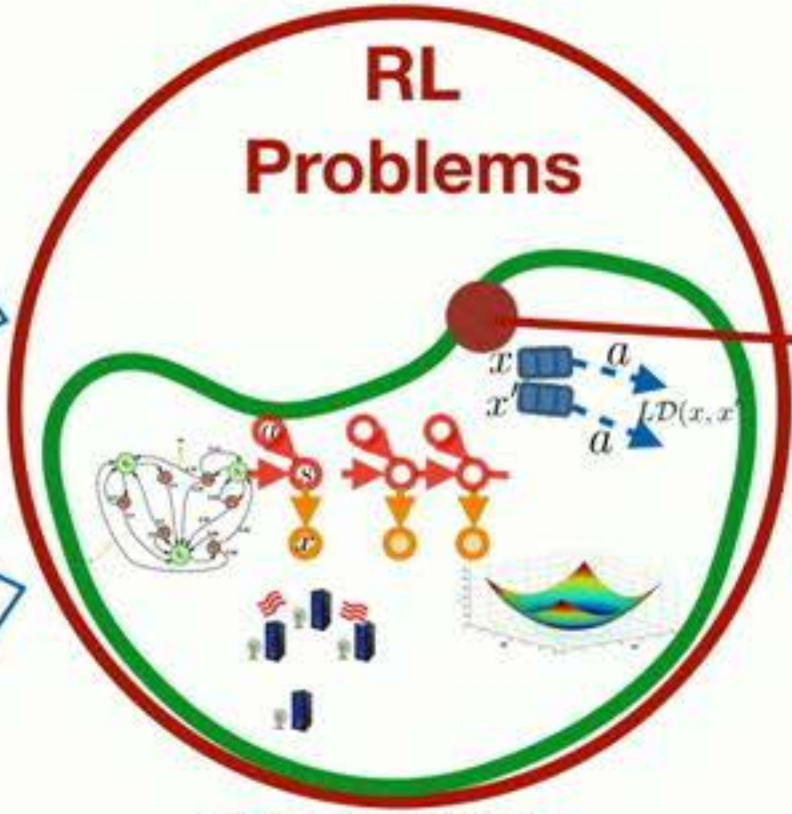
Online Policy Evaluation

Reduction from Policy Evaluation to No-Regret Online Learning

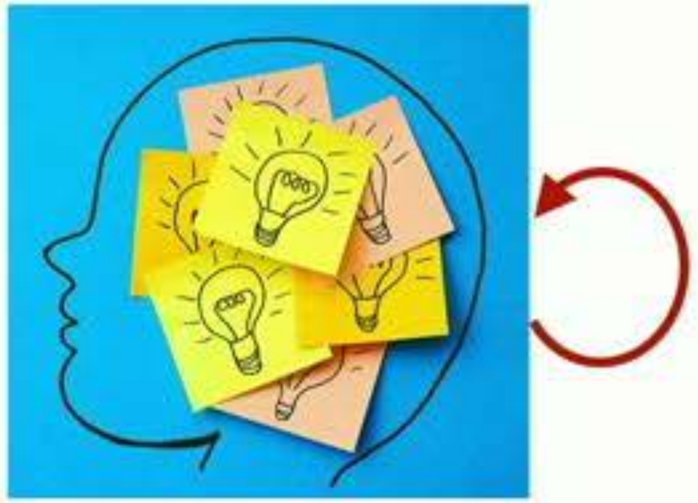
[Sun, Bagnell, UAI 15, Best Student Paper]

Imitation

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



[Sun et.al., 18, arXiv]

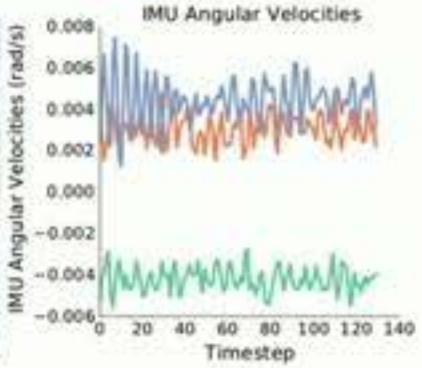


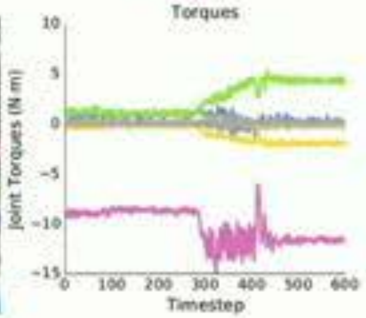
Memory

System ID

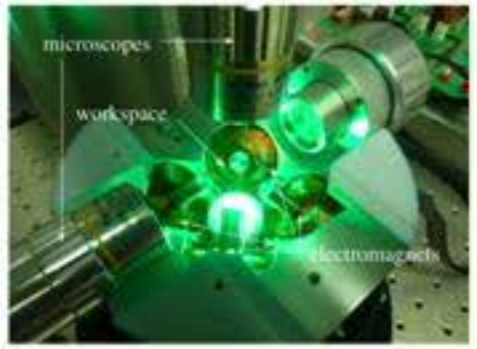
Predictive State Inference Machines

[Sun et.al., ICML 16; Venkatraman & Sun et.al., IJCAI 16, Sun et.al., ICRA 14]





(b) Kinova Mico Grasping



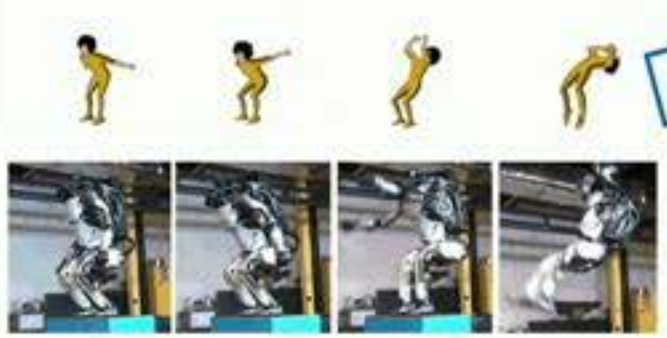
Online Policy Evaluation

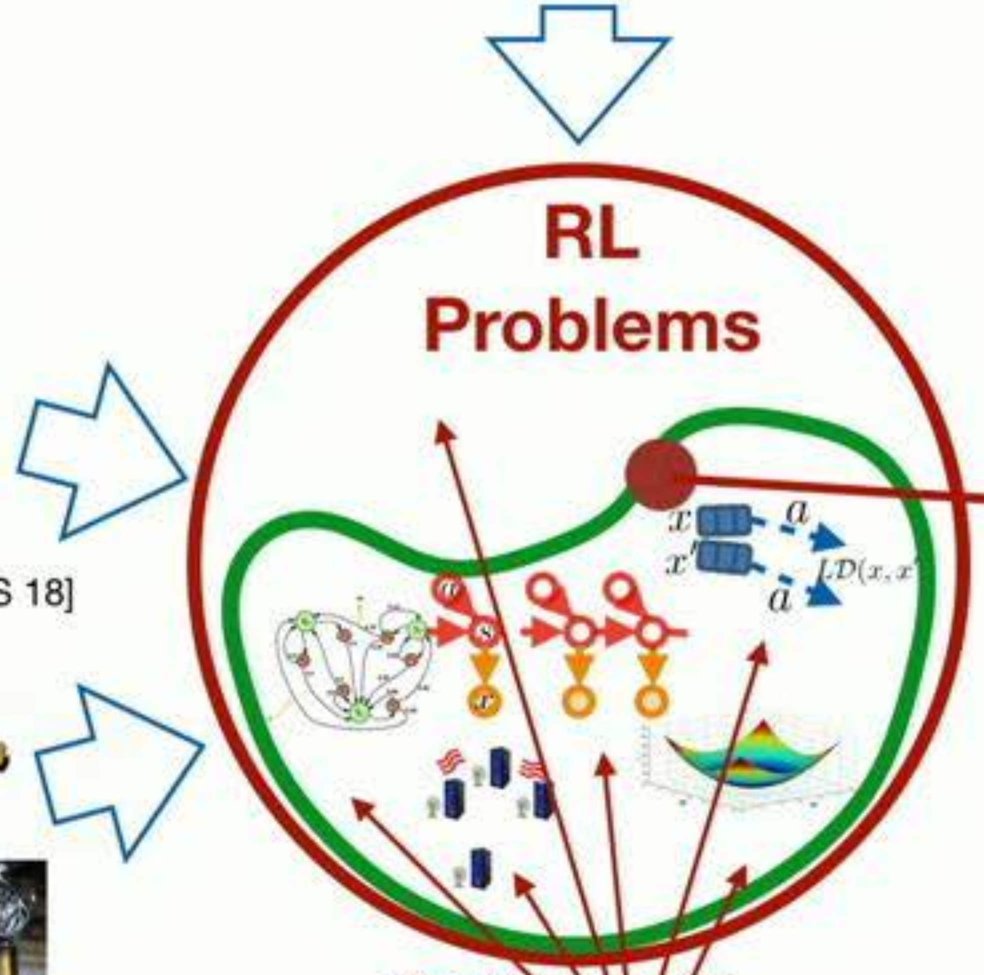
Reduction from Policy Evaluation to No-Regret Online Learning

[Sun, Bagnell, UAI 15, Best Student Paper]

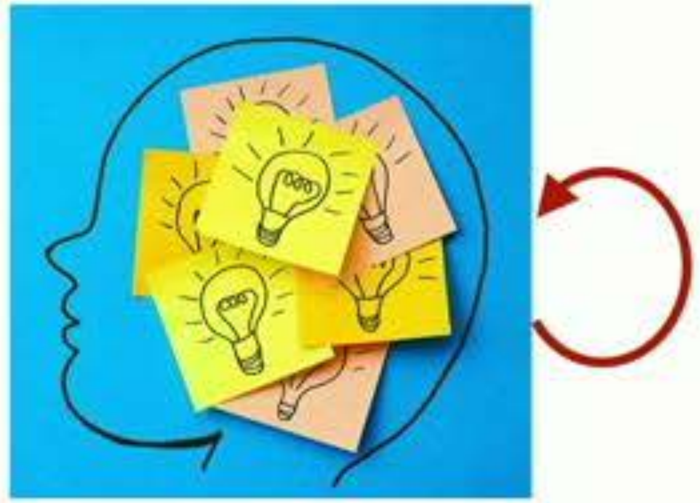
Imitation

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





[Sun et.al., 18, arXiv]

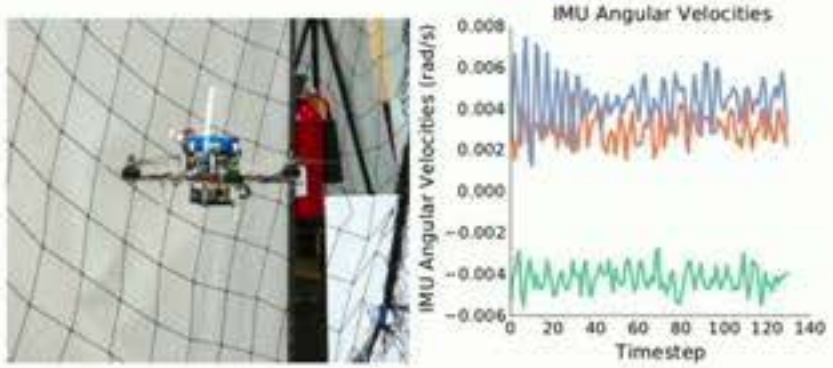


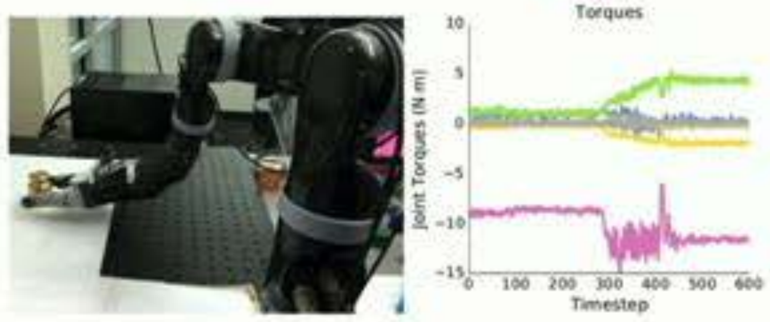
Memory

System ID

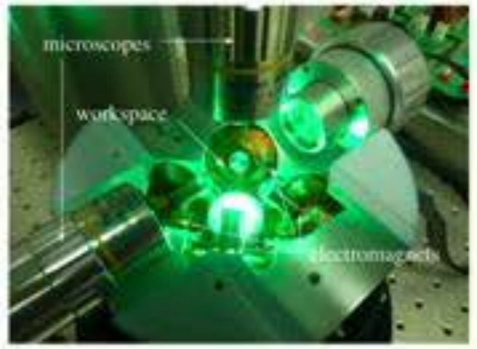
Predictive State Inference Machines

[Sun et.al., ICML 16; Venkatraman & Sun et.al., IJCAI 16, Sun et.al., ICRA 14]





(b) Kinova Mico Grasping



Thank You

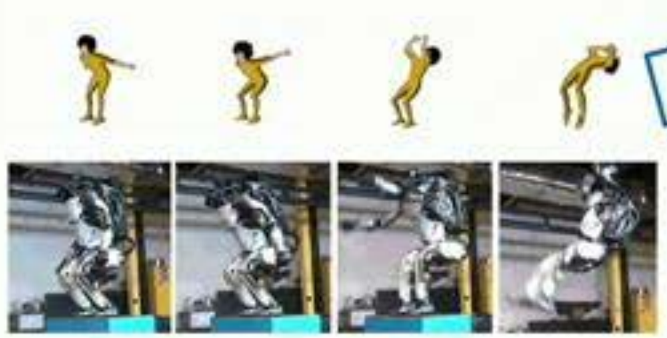
Online Policy Evaluation

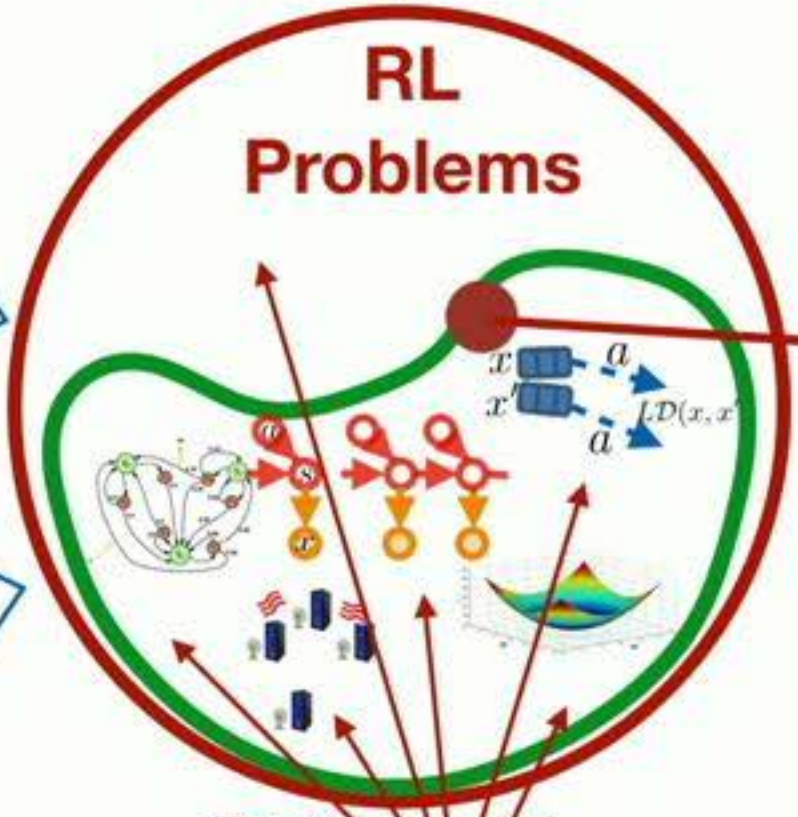
Reduction from Policy Evaluation to No-Regret Online Learning

[Sun, Bagnell, UAI 15, Best Student Paper]

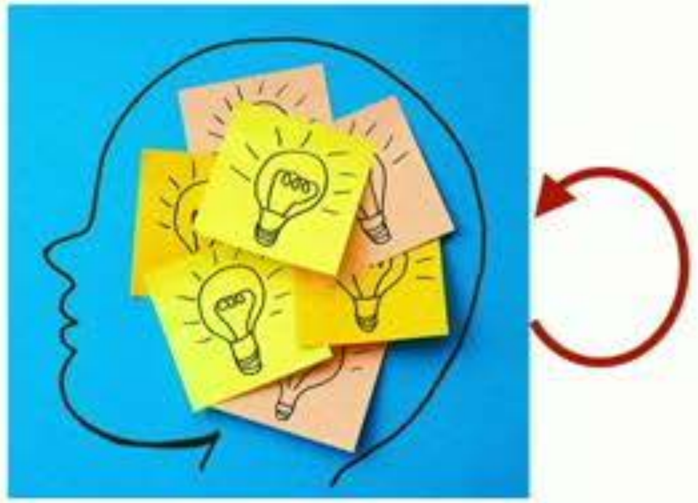
Imitation

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





[Sun et.al., 18, arXiv]

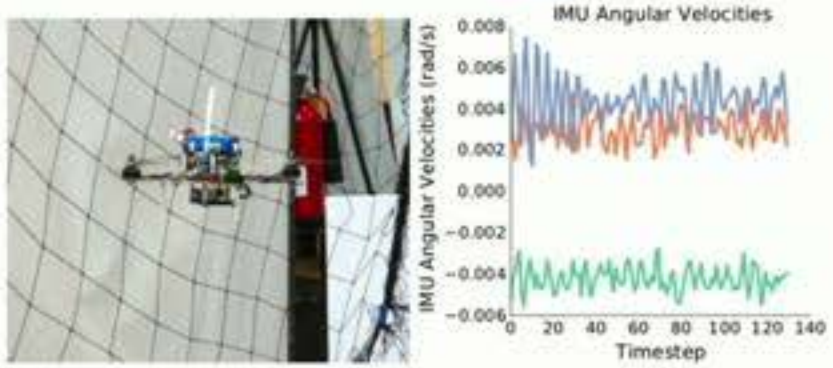


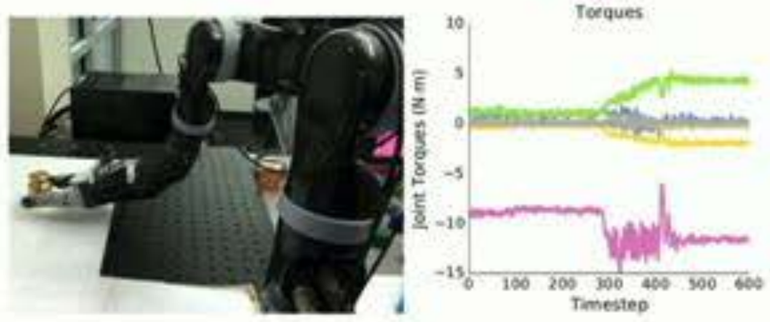
Memory

System ID

Predictive State Inference Machines

[Sun et.al., ICML 16; Venkatraman & Sun et.al., IJCAI 16, Sun et.al., ICRA 14]





(b) Kinova Mico Grasping