CORL: A Continuous-state Offset-dynamics Reinforcement Learner

Emma Brunskill, Bethany R. Leffler, Lihong Li, Michael L. Littman and

Nicholas Roy

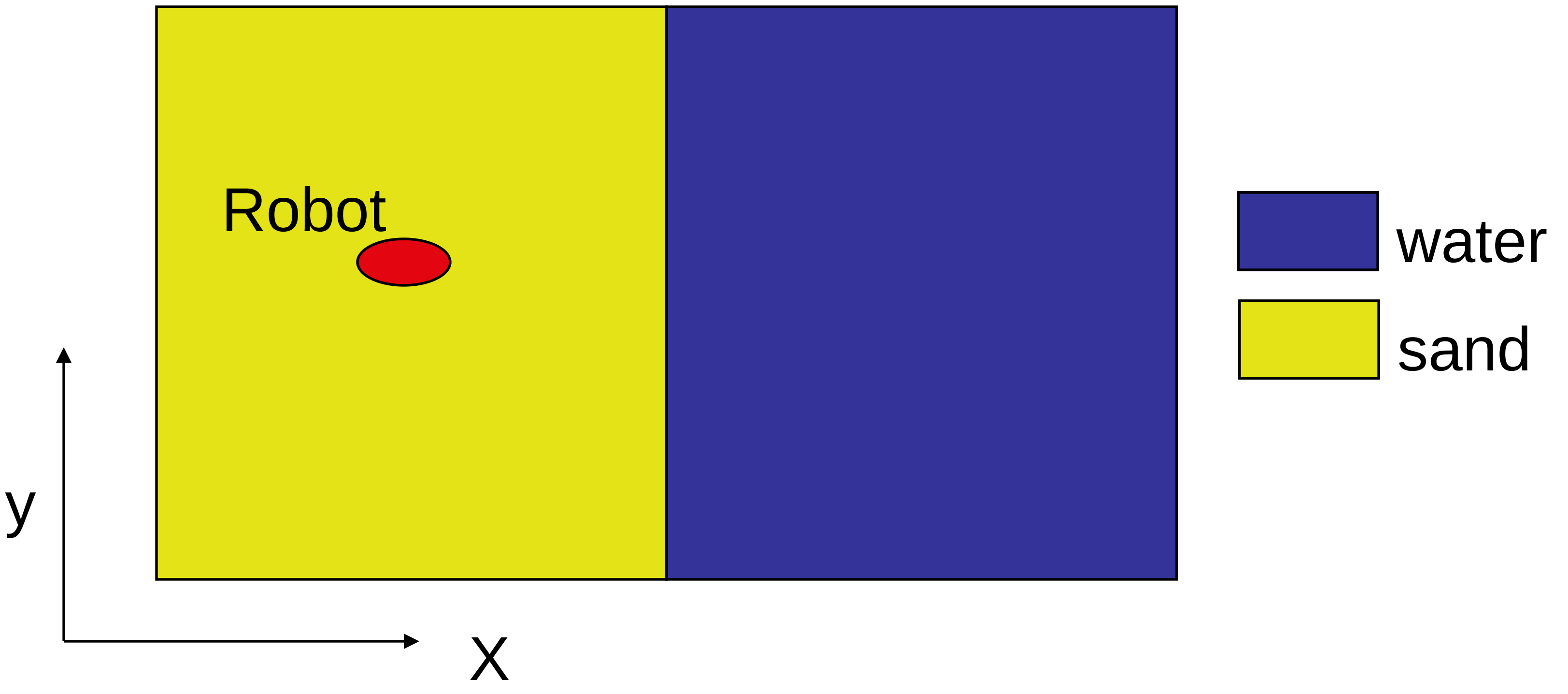
Massachusetts Institute of Technology

Rutgers University

July 2008

Motivation

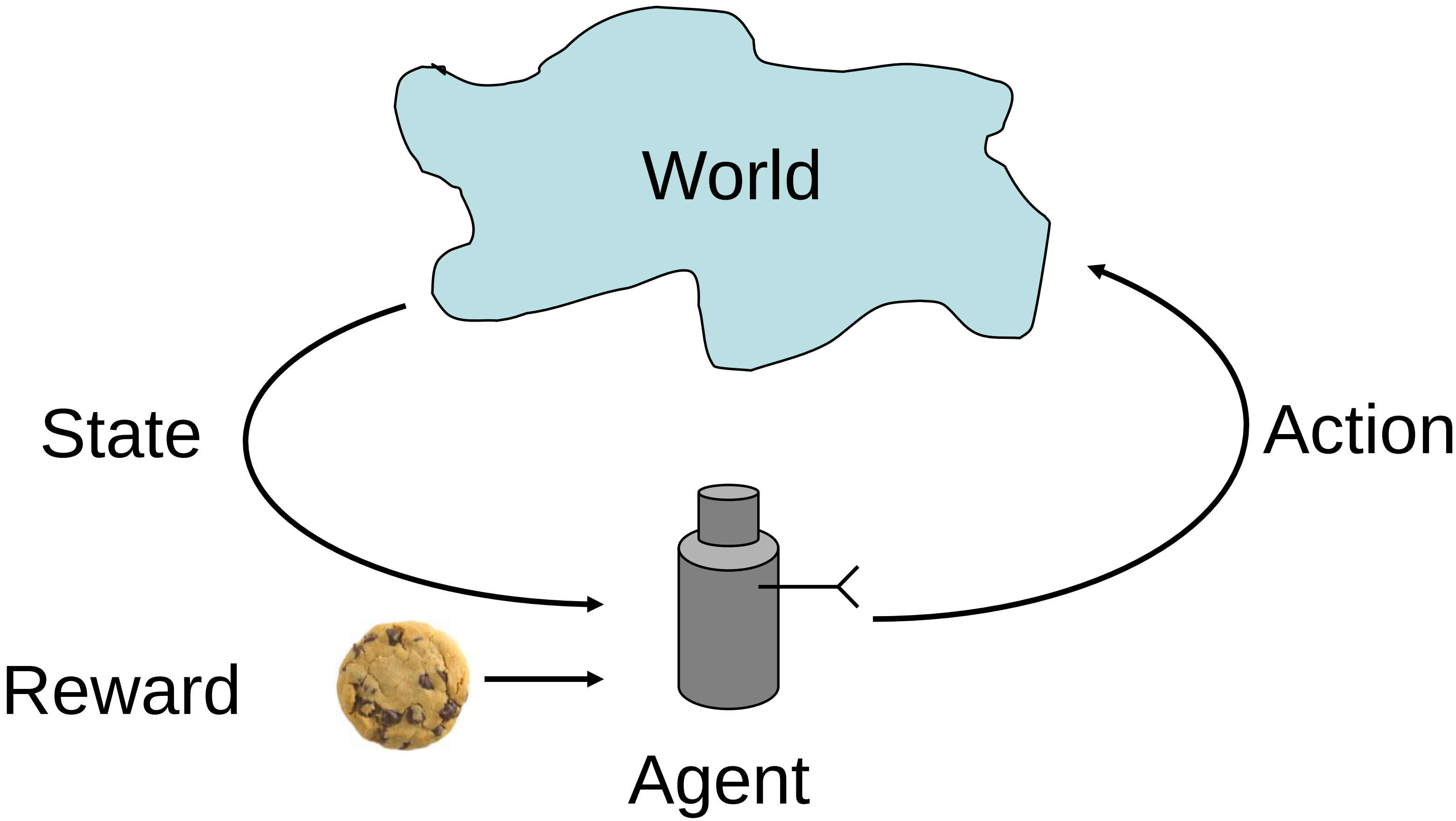
- Solve large reinforcement learning (RL) problems
- Leverage world structure for faster learning



Contributions

- RL algorithm for typed continuous domains with noisy offset dynamics
- Prove amount of experience needed scales *polynomially* with state space dimension
- Explicitly consider error due to approximate planning
- Robot experiment shows dynamics model approximation adequate for real world learning

Reinforcement Learning



Goal: Maximize expected sum of future rewards

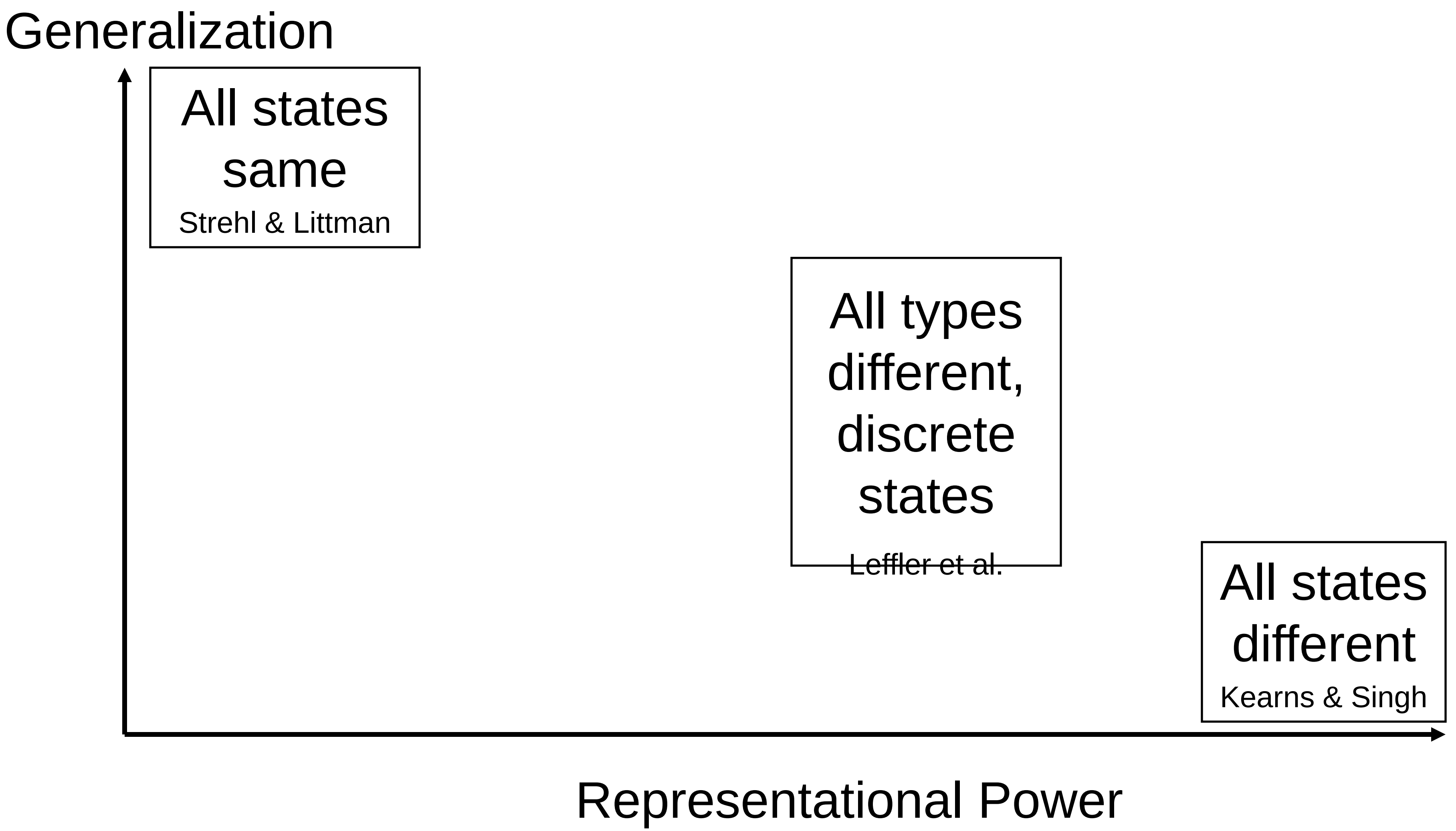
World Representation

vs

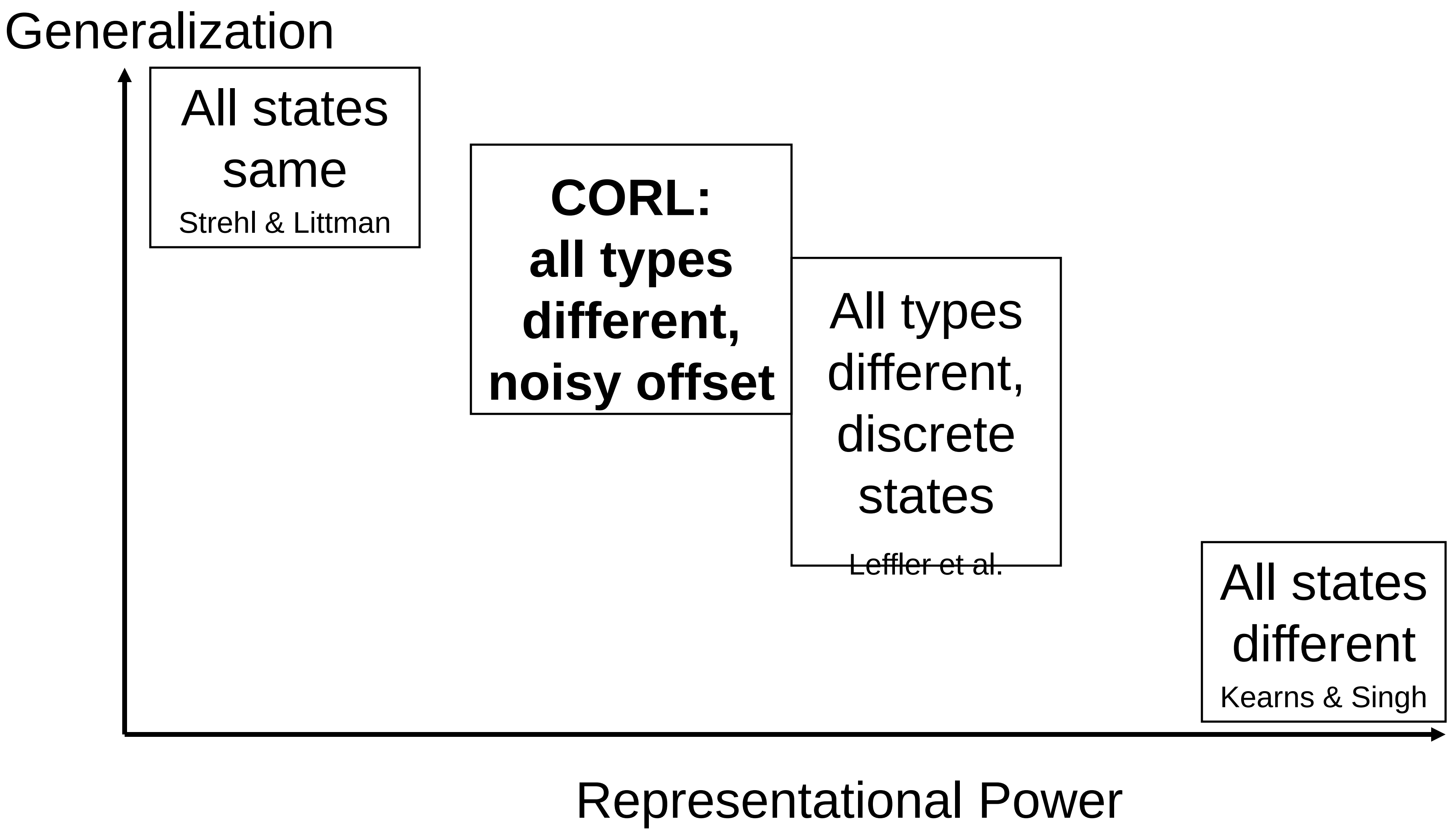
(12.3752,13.8763)

Grid Cell 17 or
At(*Robot*,*Kitchen*)

Dynamics Representation

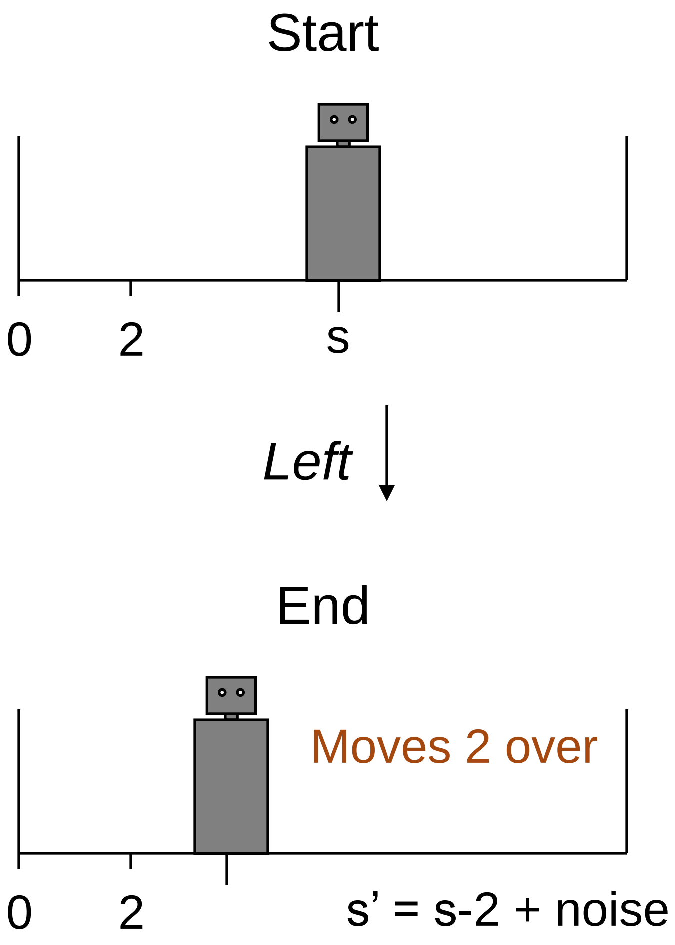


Dynamics Representation



CORL Dynamics

- Continuous-valued states S
- Finite set of types M
- Known function *Type*: $S \rightarrow M$
- Noisy offset typed dynamics



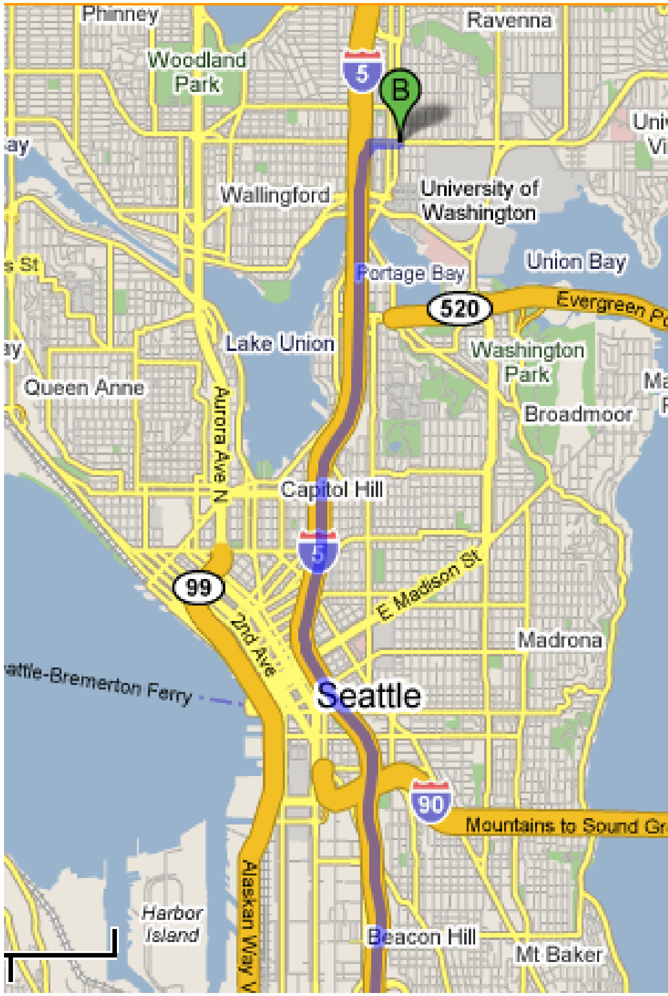
CORL Offset-Typed Dynamics

$$s' = s + \boxed{\beta_{m,a}} + \varepsilon$$

$$m = Type(s)$$

$$\varepsilon \sim N(0, \boxed{\Sigma_{m,a}})$$

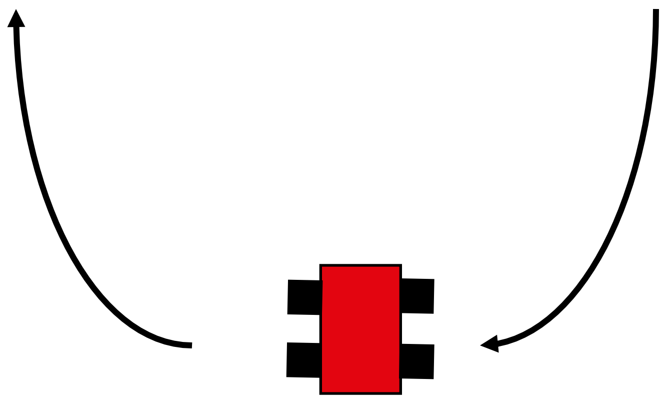
Learn



(Dudek et al.'s AQUA robot http://www.aquarobot.net:8080/AQUA)/; http://maps.google.com

Model-based RL

Think hard: estimate models & plan



Act in world

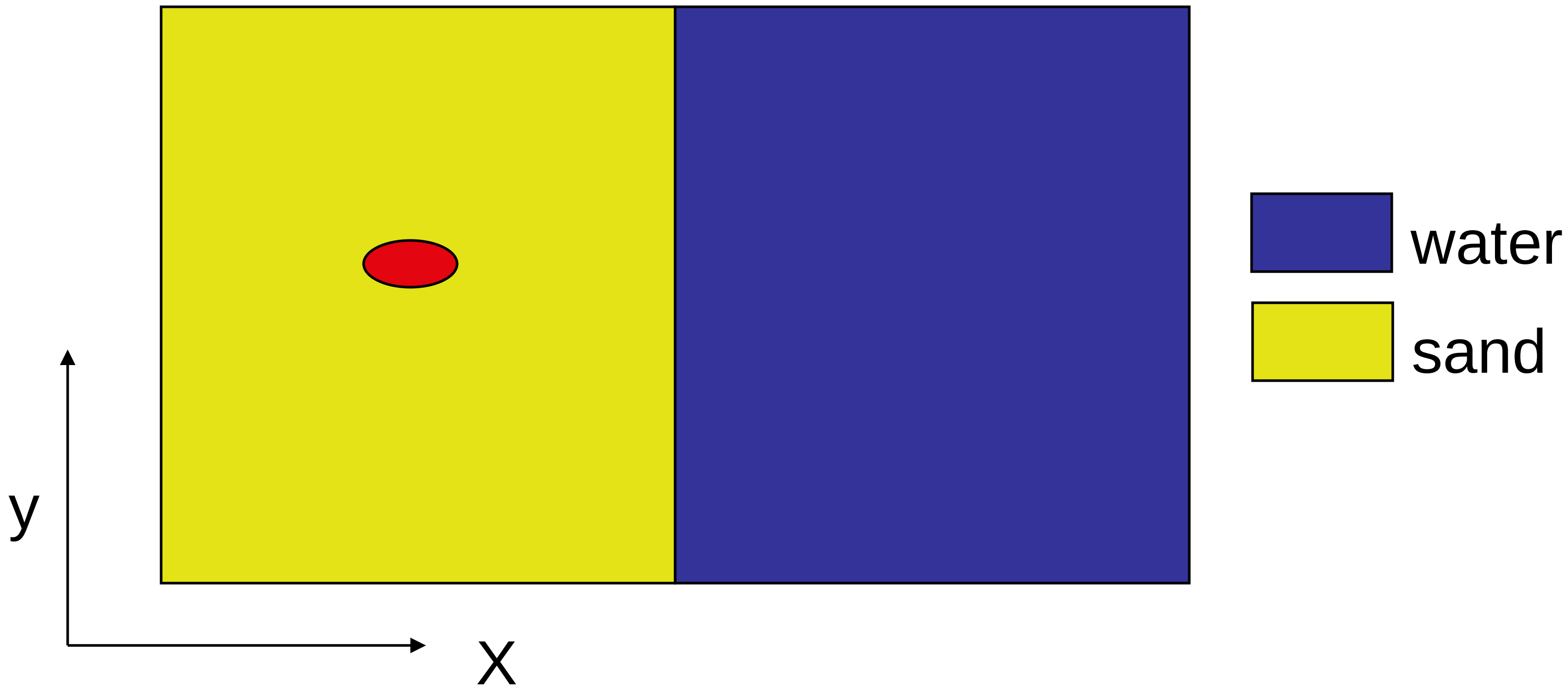
Balancing Exploration with Exploitation

Probably Approximately Correct (PAC): learn quickly but do not require optimality during learning

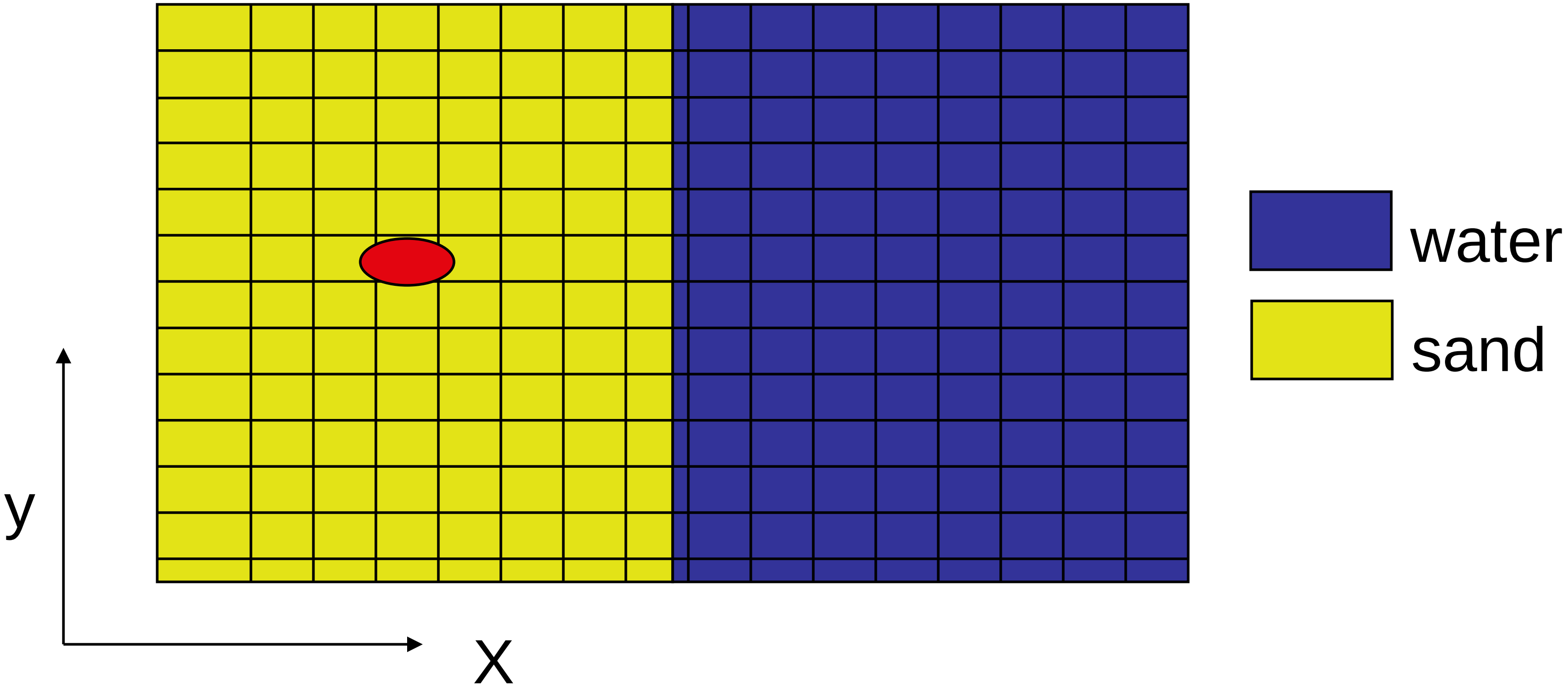
PAC RL Approaches

- *Discrete states & actions*: E^3 (Kearns and Singh 1998), R-max (Brafman & Tenneholtz 2002)
- *Discrete typed offset*: RAM-Rmax (Leffler et al. 2007)
- *Continuous linear dynamics with known variance*: (Strehl & Littman 2008)
- *Continuous typed offset with unknown variance*: CORL

Small Example



Small Example



R-max Algorithm: Initialize

Known/ Unknown

	S1	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

Reward

	S1	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	R_{max}	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

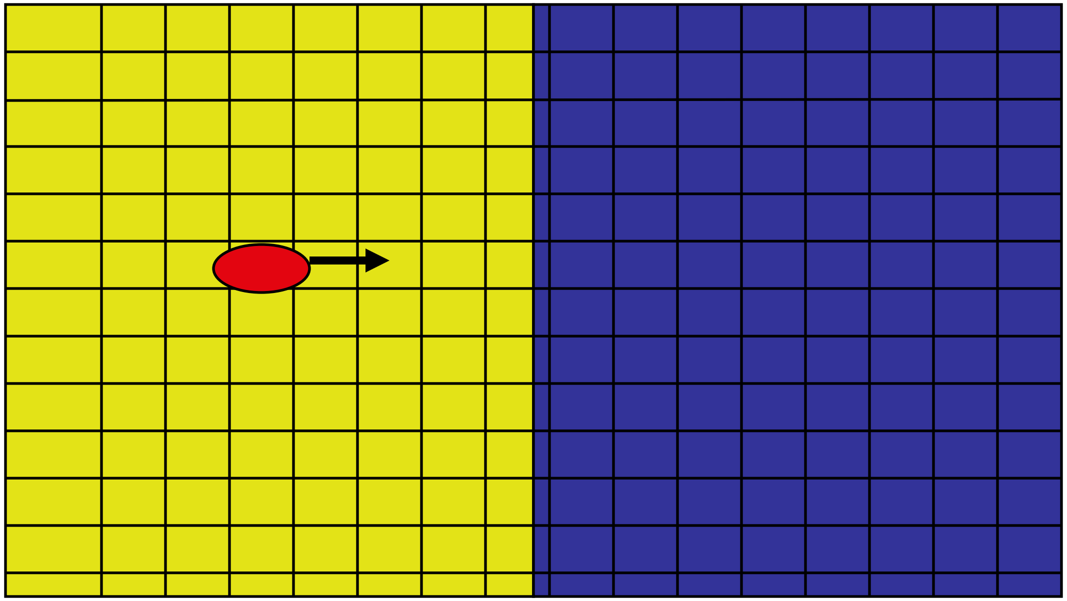
Transition Counts

	S1	S2	S3	S4	…
↑	0	0	0	0	
→	0	0	0	0	
↓	0	0	0	0	
←	0	0	0	0	

Create “known” MDP

R-max: Solve & Act

- Solve “known” MDP
- Take best action



R-max Algorithm: Update

Reward

Known/ Unknown

	S2	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

	S2	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	R_{max}	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S2	S2	S3	S4	…
↑	0	0	0	0	
→	0	0	**1**	0	
↓	0	0	0	0	
←	0	0	0	0	

Increment counts for state-action tuple

R-max Algorithm: Update Cont.

Known/ Unknown

	S2	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	**K**	U	
↓	U	U	U	U	
←	U	U	U	U	

Reward

	S2	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	**R**	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S2	S2	S3	S4	…
↑	3	3	4	3	
→	2	4	**5**	0	
↓	4	0	4	4	
←	2	2	4	1	

If counts for (s,a) > N, use estimated models for (s,a) when planning

R-max Algorithm

Solve known MDP

R-max Algorithm

Act using solution

Solve known MDP

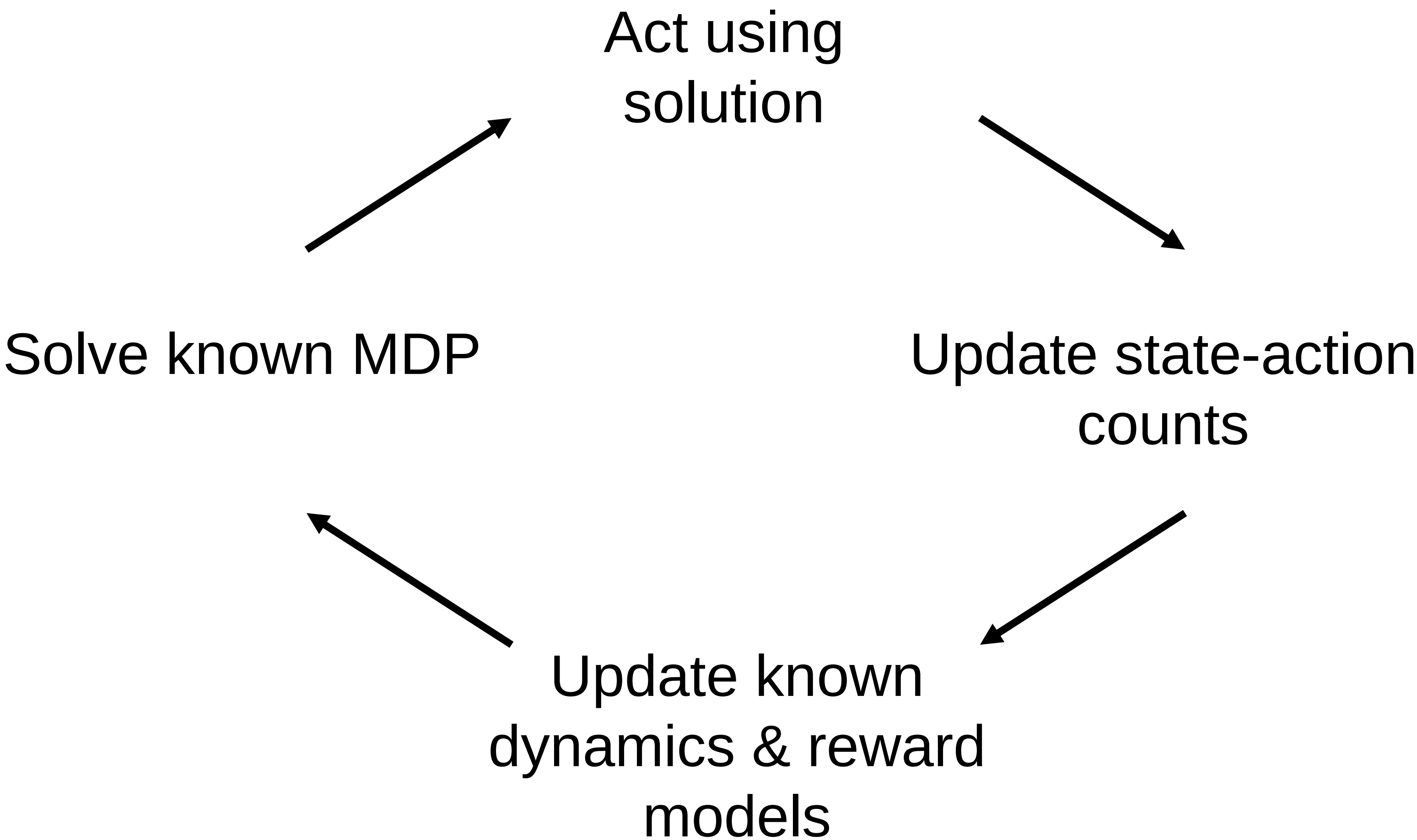
R-max Algorithm

Act using solution

Solve known MDP

Update state-action counts

R-max Algorithm



CORL: Initialization

Known/
Unknown

↑	U	U
→	U	U
↓	U	U
←	U	U

Transition
Counts

↑	0	0
→	0	0
↓	0	0
←	0	0

Reward

Types

↑	R_{max}	R_{max}
→	R_{max}	R_{max}
↓	R_{max}	R_{max}
←	R_{max}	R_{max}

CORL Algorithm: Solve

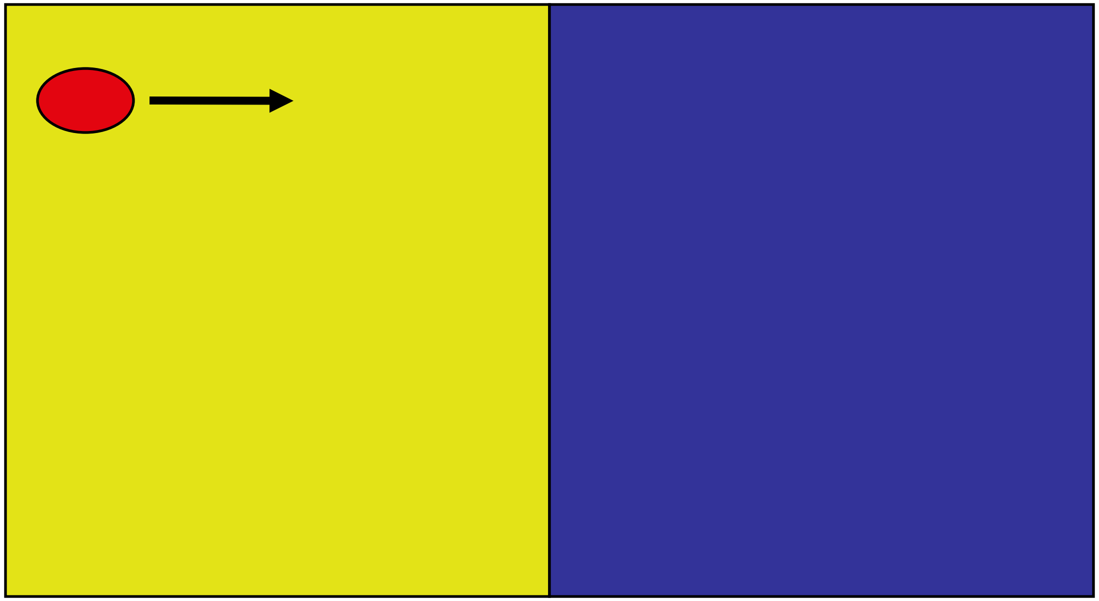
- No exact planner for general continuous-state MDPs
- Use Fitted Value Iteration to approximately solve

CORL Algorithm: Solve

- No exact planner for general continuous-state MDPs
- Use Fitted Value Iteration to approximately solve
  - Perform Bellman backups at a finite set of states
  - Approximate value at other states using function approximation
- Consider this error in sample complexity bounds

CORL Algorithm: Act

Take best action given approximate solution



CORL: Update Counts

Known/
Unknown

↑	U	U
→	U	U
↓	U	U
←	U	U

Reward

Types

↑	R_{max}	R_{max}
→	R_{max}	R_{max}
↓	R_{max}	R_{max}
←	R_{max}	R_{max}

Transition
Counts

↑	0	0
→	1	0
↓	0	0
←	0	0

Increment counts for
type-action tuple

CORL: Label Known Tuples

Known/
Unknown

↑	U	U
→	**K**	U
↓	U	U
←	U	U

Transition
Counts

↑	1	3
→	**5**	2
↓	4	4
←	2	3

Reward

Types

↑	R_{max}	R_{max}
→	**R**	R_{max}
↓	R_{max}	R_{max}
←	R_{max}	R_{max}

Label state-type as known when counts exceed threshold

CORL: Estimate Dynamics

For known type-action tuples estimate dynamics parameters from experience

$$\tilde{\beta}_{at} = \frac{\sum_{i=1}^{counts_{at}} (s'_i - s_i)}{counts_{at}}$$

$$\tilde{\Sigma}_{at} = \frac{\sum_{i=1}^{counts_{at}} (s'_i - s_i - \tilde{\beta}_{at})(s'_i - s_i - \tilde{\beta}_{at})^T}{counts_{at}}$$

CORL Algorithm

Act using approximate solution

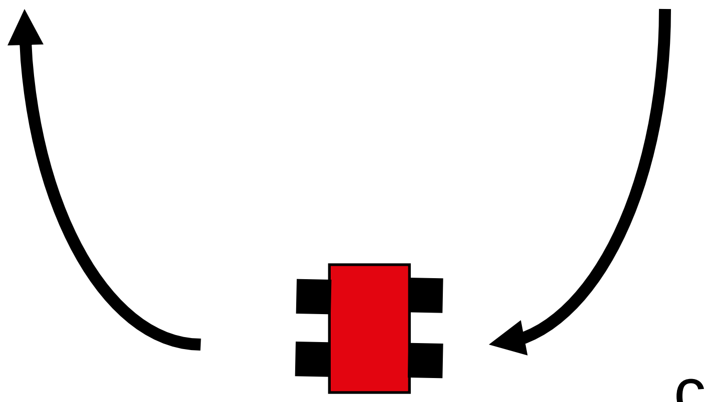
Update **type-action** counts

Update known dynamics & reward models

Approximately solve known MDP

Complexity

Think hard: estimate models & plan



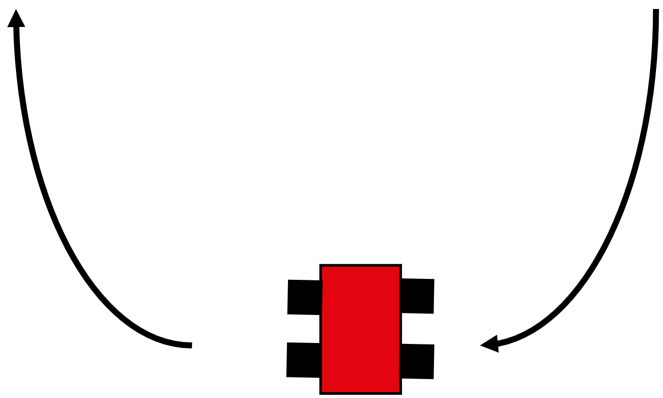
Sample complexity

Act in world

Complexity

Computational complexity

Think hard: estimate models & plan



Act in world

Theoretical Results

- Estimate offset dynamics parameters
- For diagonal covariance, approximate model can be used to get near-optimal behavior
- Bound error due to approximate planning (Chow and Tsitsiklis 1991)
- Combine ideas to bound sample complexity

CORL Theorem

Assuming

- a continuous-state noisy offset dynamics MDP with diagonal covariances

Given

- δ and ε
- |M| types
- Variance along each dimension of all the dynamics models bounded by $\left[\sigma_{\min}^2, B_\sigma^2\right]$
- Each offset parameter bounded by $\left|\beta_i\right| < B_\beta$

CORL Theorem

Then on all but N_{total} timesteps CORL will follow a 4ε-optimal policy with probability at least $1-2\delta$, where

$$N_{total} = poly\left(N_{\dim}, |A|, |M|, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-\gamma}, \frac{1}{\sigma_{\min}}, B_{\beta}, B_{\sigma} \right)$$

Sample Complexity Results

R-max	$\tilde{O}(\lvert A\rvert\lvert S\rvert^2)$
RAM-Rmax	$\tilde{O}(\lvert A\rvert\lvert M\rvert\lvert S\rvert)$
CORL	$\tilde{O}(\lvert A\rvert\lvert M\rvert S_{dim}^2)$

|S| = size of the state space,

|A| = number of actions,

|M| = number of types,

N_{dim} = dimensionality of the state space

Sample Complexity Summary

R-max	*Exponential* with state space dimension*
RAM-Rmax	
CORL	*Polynomial* with state space dimension

*Using uniform grid-based discretization

→ Result: significantly less experience needed to perform well

Experimental Motivation

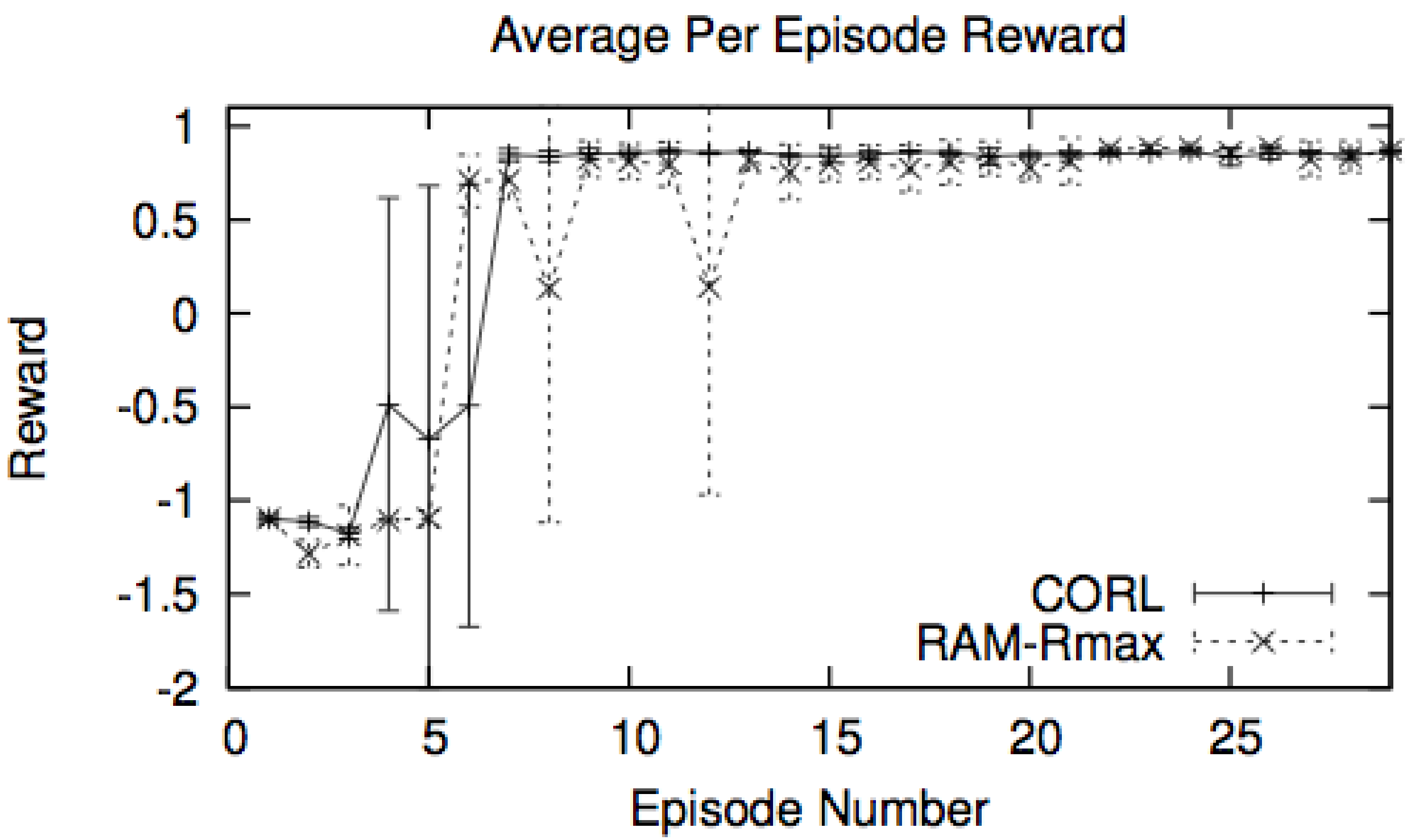
- Examine if dynamics model is sufficient to enable good performance in a real world task

Navigation over Varying Terrain

Movie

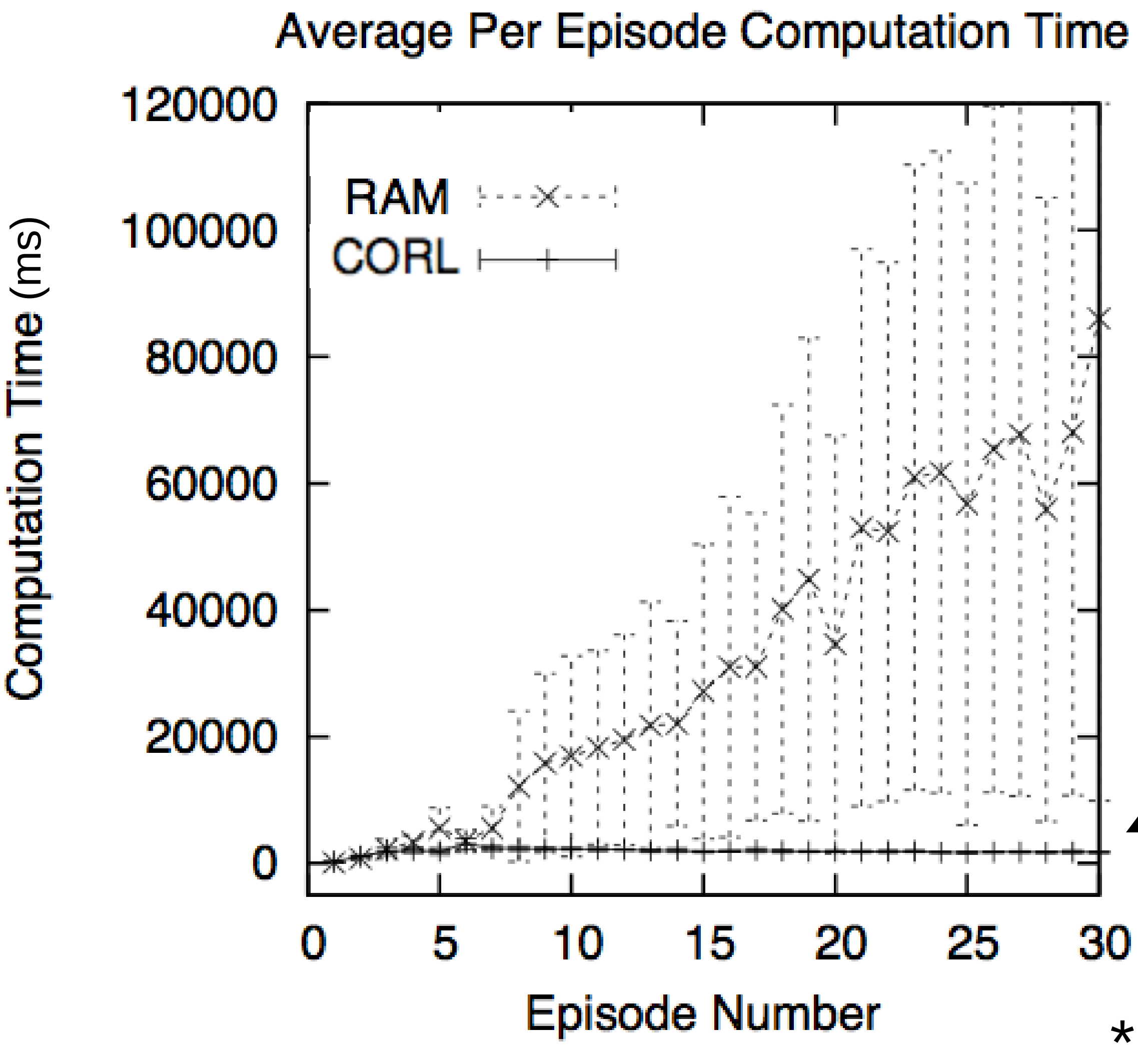
QuickTime™ and a
H.264/AVC decompressor
are needed to see this picture.

Generalization → Fast Learning



*Averaged over 3 runs

Computational Cost



Compact representation keeps computational cost flat

*Averaged over 3 runs

Open Issues

- Faster continuous-state MDP planning
- Experimental results on other domains
- Consider gap between theory and experiment

CORL Summary

- RL algorithm that brings idea of types to continuous-valued representation
- Enables faster learning
  - Amount of experience needed scales *polynomially* with dimension of state space (instead of *exponentially*)
- Bound includes approximate learning error
- Robot experiment shows dynamics model approximation is adequate for good performance