

Provably Efficient RL via Latent State Decoding



Simon S. Du



**Akshay
Krishnamurthy**



Nan Jiang



**Alekh
Agarwal**



Miro Dudík

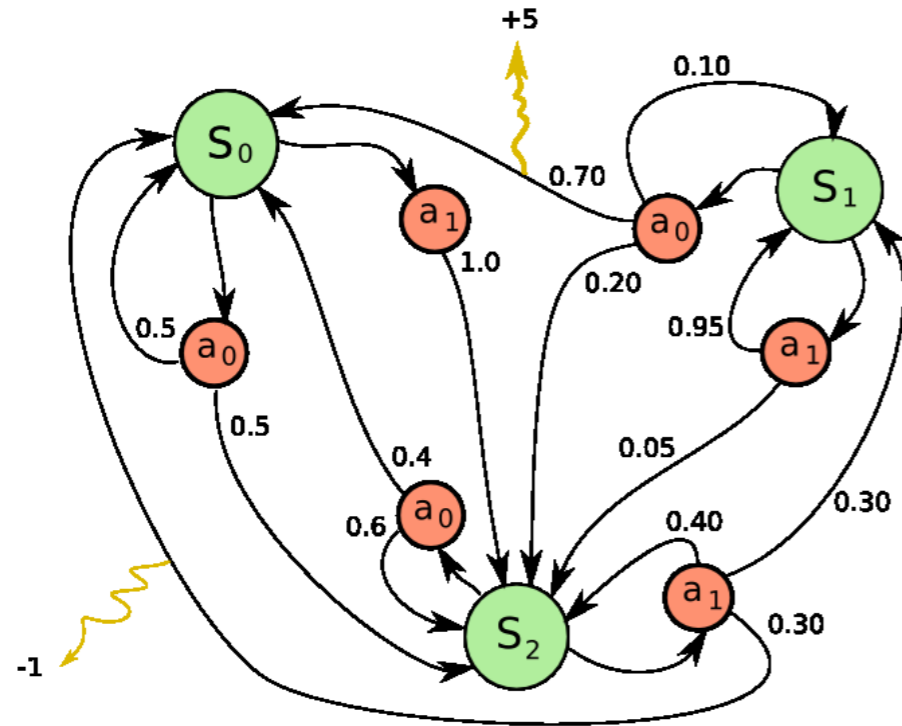


**John
Langford**

RL theory vs practice



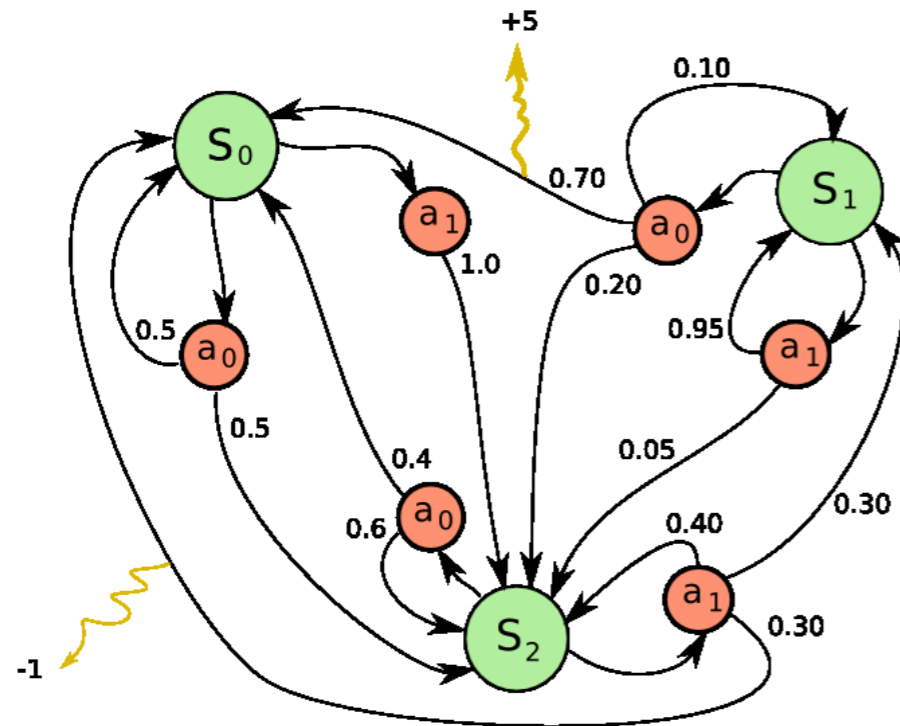
RL theory vs practice



Theory

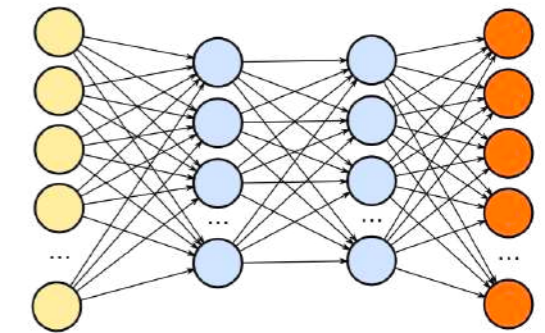
Simple tabular environments
No generalization

RL theory vs practice



Theory

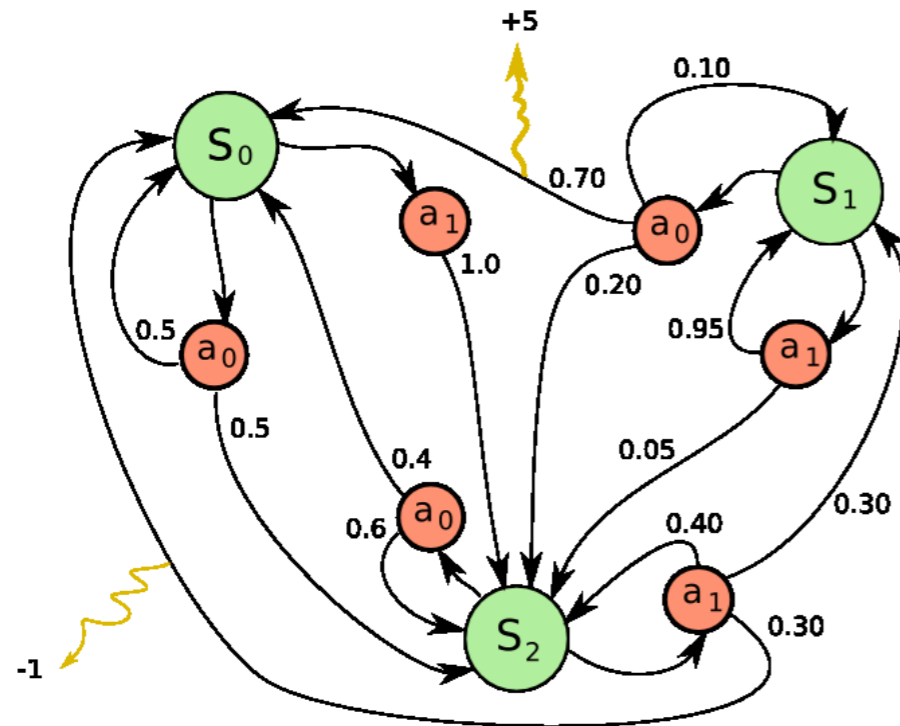
Simple tabular environments
No generalization



Practice

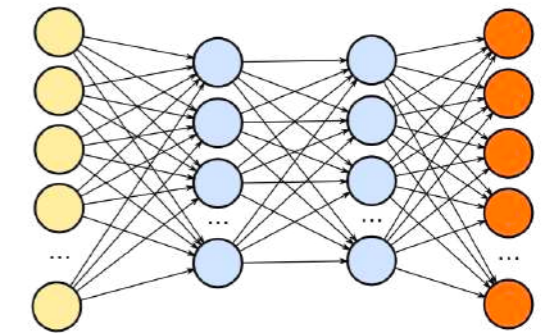
Complex rich-observation environments
Generalization via function approximation

RL theory vs practice



Theory

Simple tabular environments
No generalization



Practice

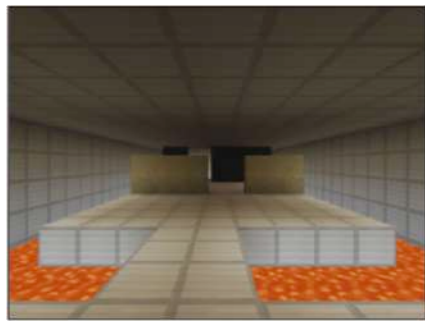
Complex rich-observation environments
Generalization via function approximation

Can we design provably sample-efficient RL algorithms for rich observation environments?

Block MDPs

A structured model for rich observation RL

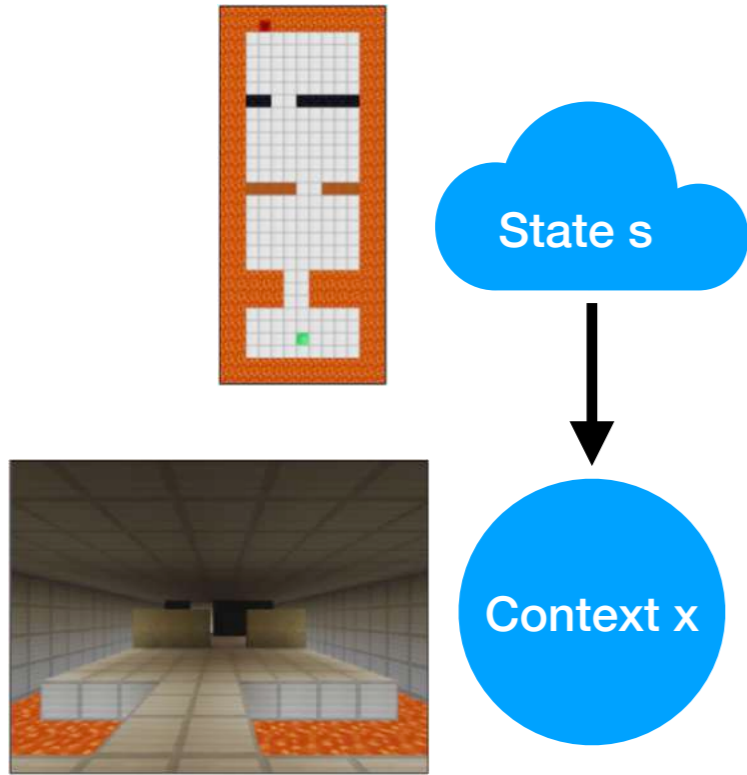
Block MDPs



A structured model for rich observation RL

- Agent only observes rich context (visual signal)

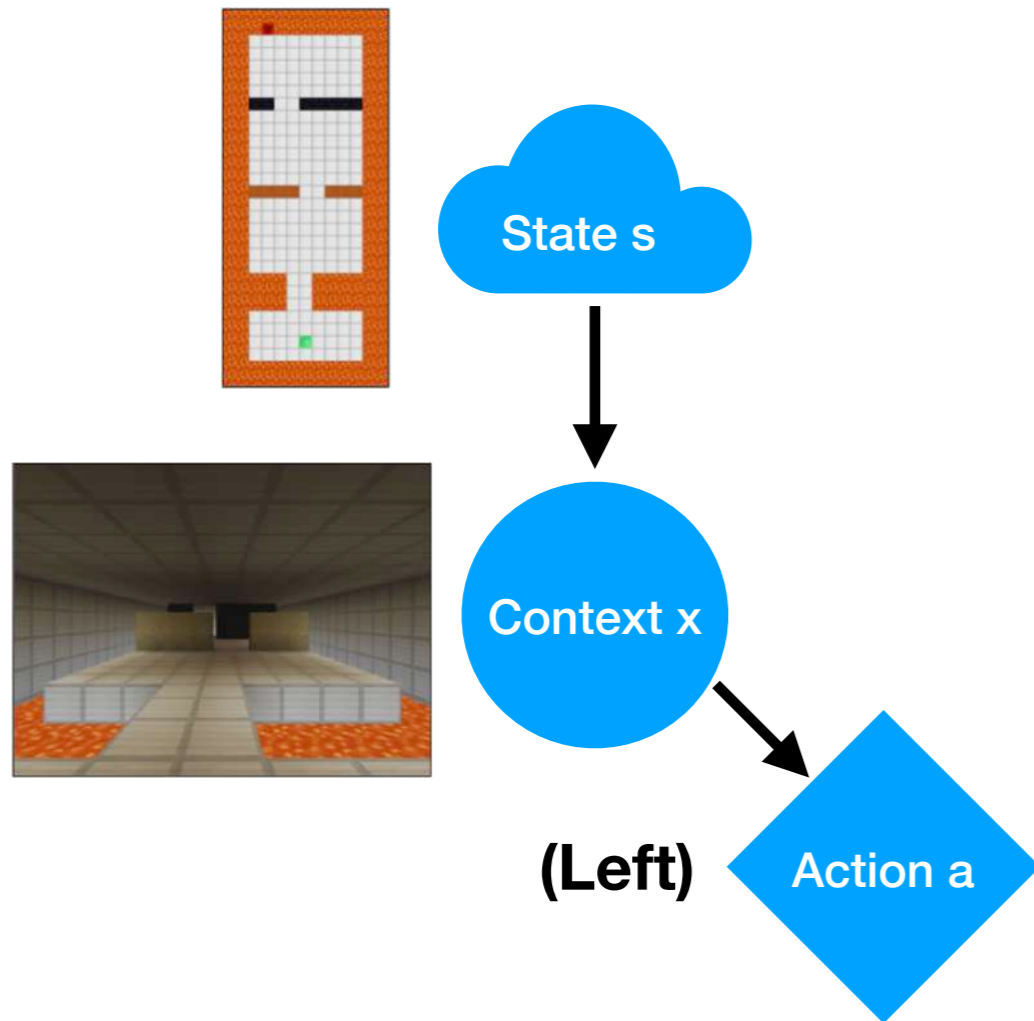
Block MDPs



A structured model for rich observation RL

- Agent only observes rich context (visual signal)
- Environment summarized by small hidden state space (agent location)

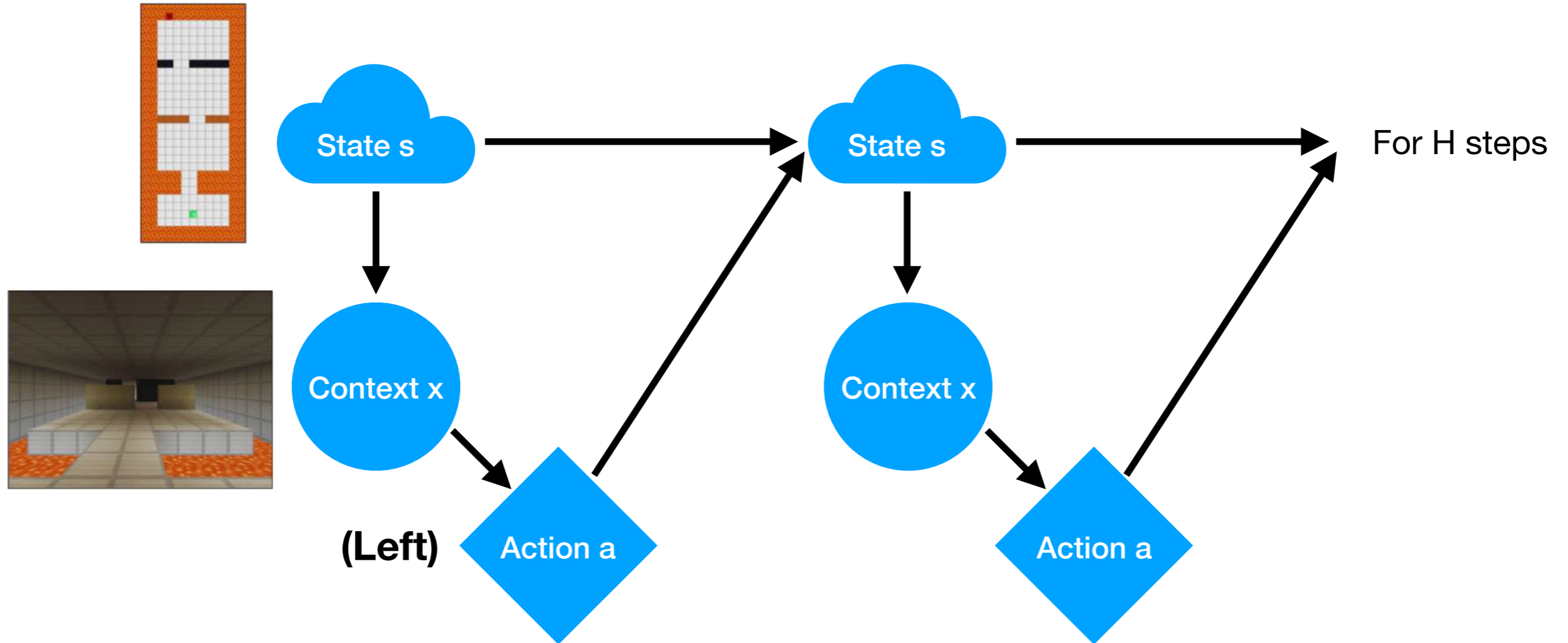
Block MDPs



A structured model for rich observation RL

- Agent only observes rich context (visual signal)
- Environment summarized by small hidden state space (agent location)

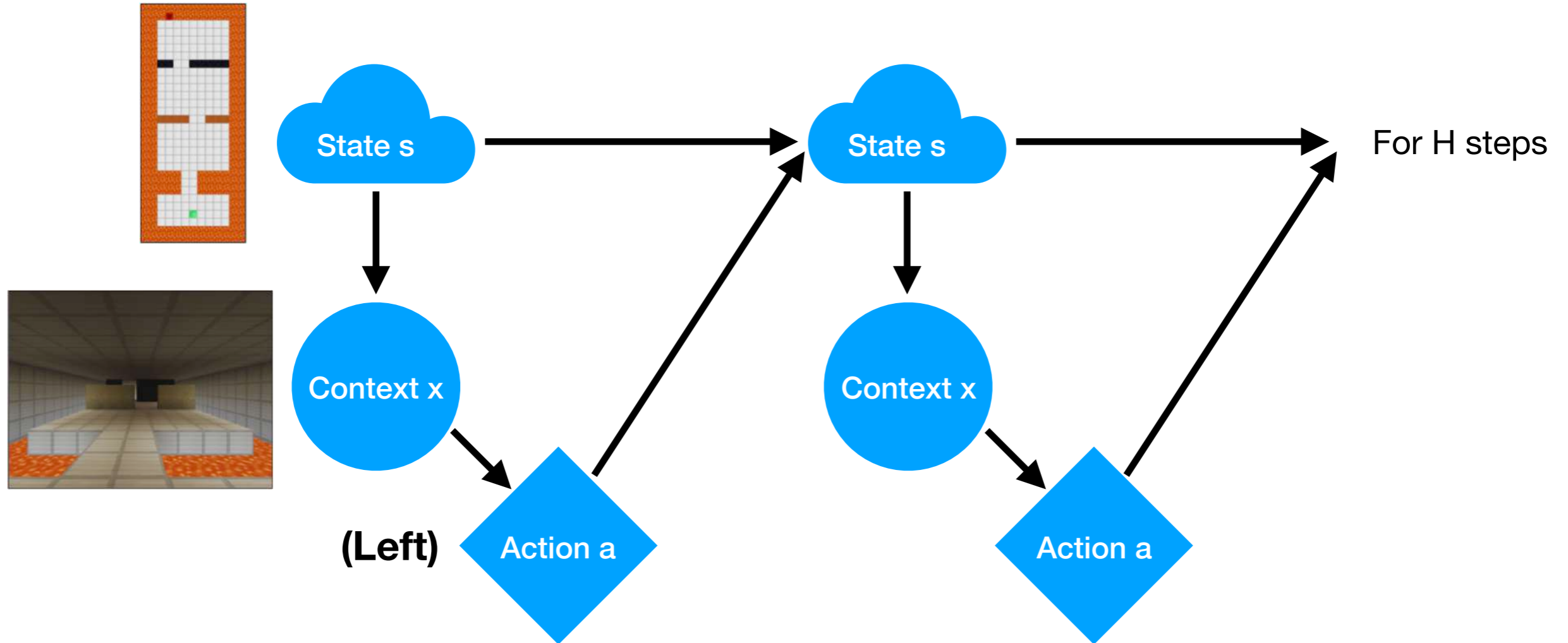
Block MDPs



A structured model for rich observation RL

- Agent only observes rich context (visual signal)
- Environment summarized by small hidden state space (agent location)

Block MDPs



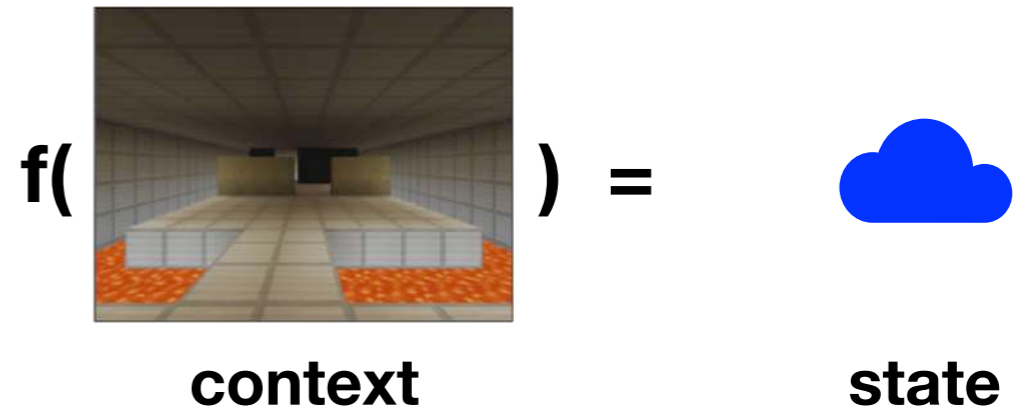
A structured model for rich observation RL

- Agent only observes rich context (visual signal)
- Environment summarized by small hidden state space (agent location)
- State can be decoded from observation

Objective: Find a Decoder



Objective: Find a Decoder

Idea: Find a function that decodes hidden states from contexts.

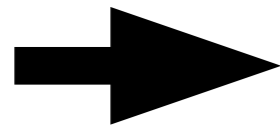


Objective: Find a Decoder

Idea: Find a function that decodes hidden states from contexts.

$$f(\text{context}) = \text{state}$$


context state

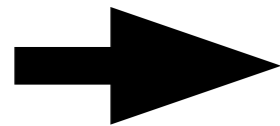
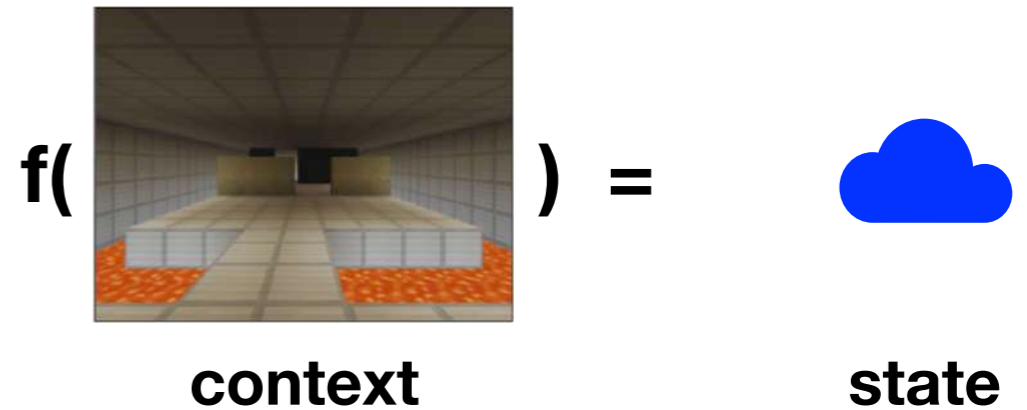


Reduce to a tabular problem

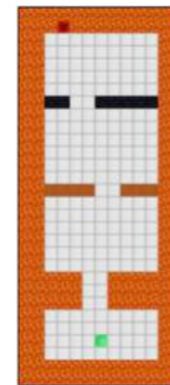


Objective: Find a Decoder

Idea: Find a function that decodes hidden states from contexts.



Reduce to a tabular problem

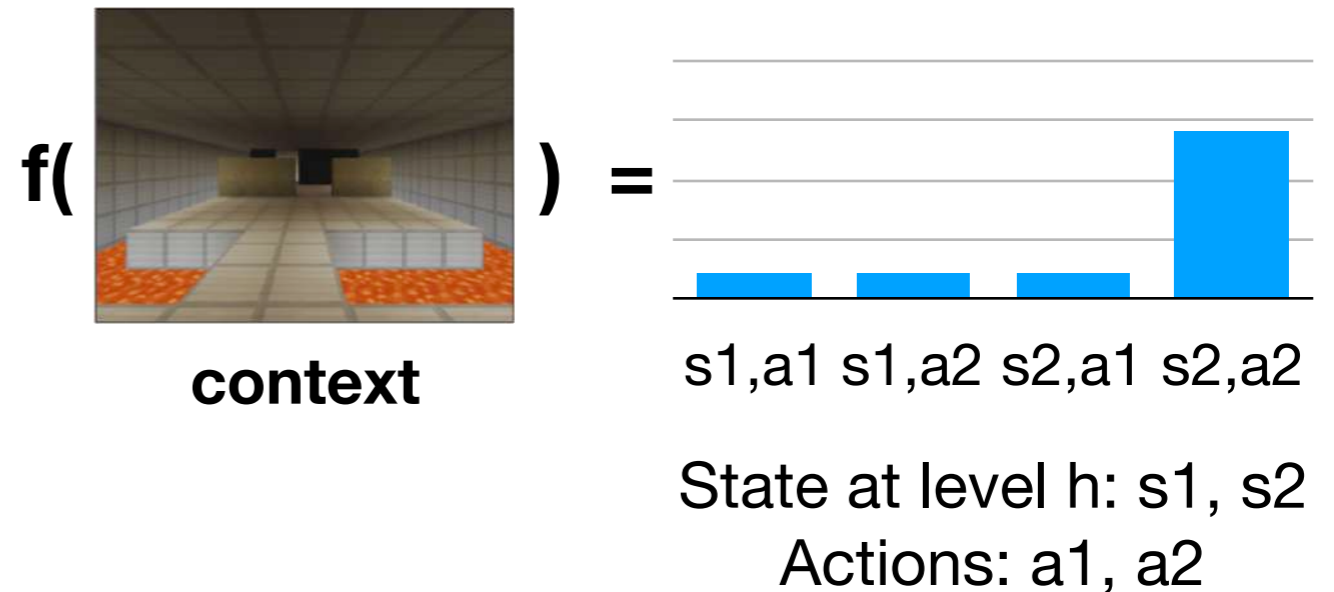


Main Challenge: There is no label (we cannot observe hidden states).

Approach

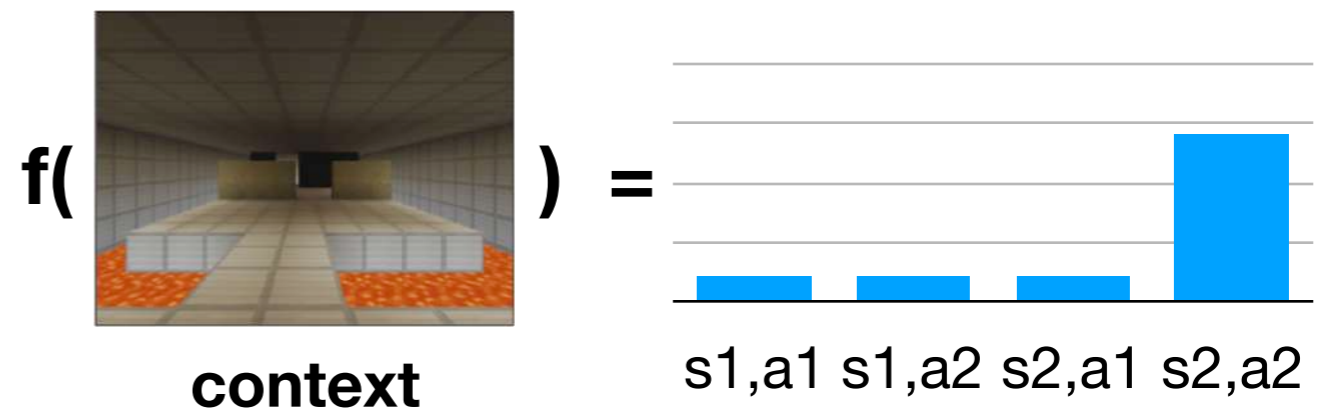
Approach

Our Approach: Learn a function that predicts the conditional probability of (previous state, action) pairs from contexts.
(assume access a regression oracle to learn this function)



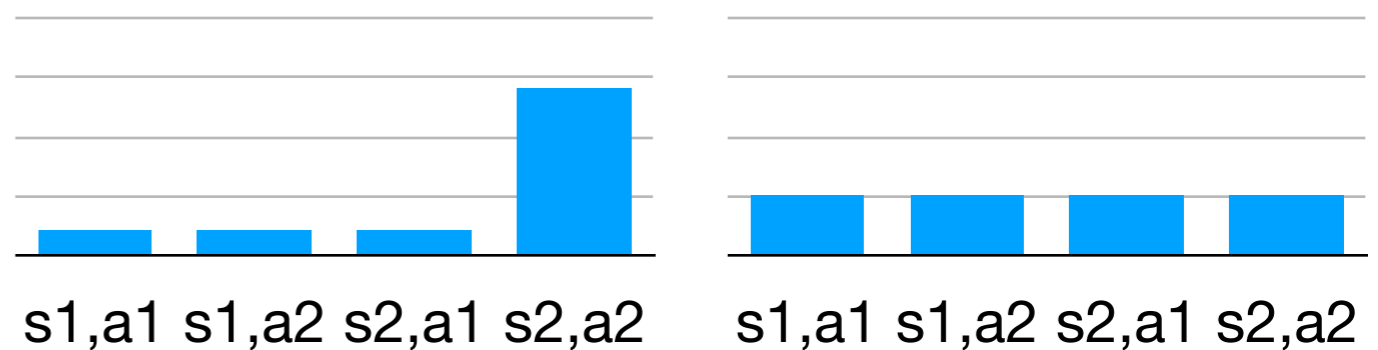
Approach

Our Approach: Learn a function that predicts the conditional probability of (previous state, action) pairs from contexts.
 (assume access a regression oracle to learn this function)



State at level h: s1, s2
 Actions: a1, a2

Different conditional probabilities correspond to different states

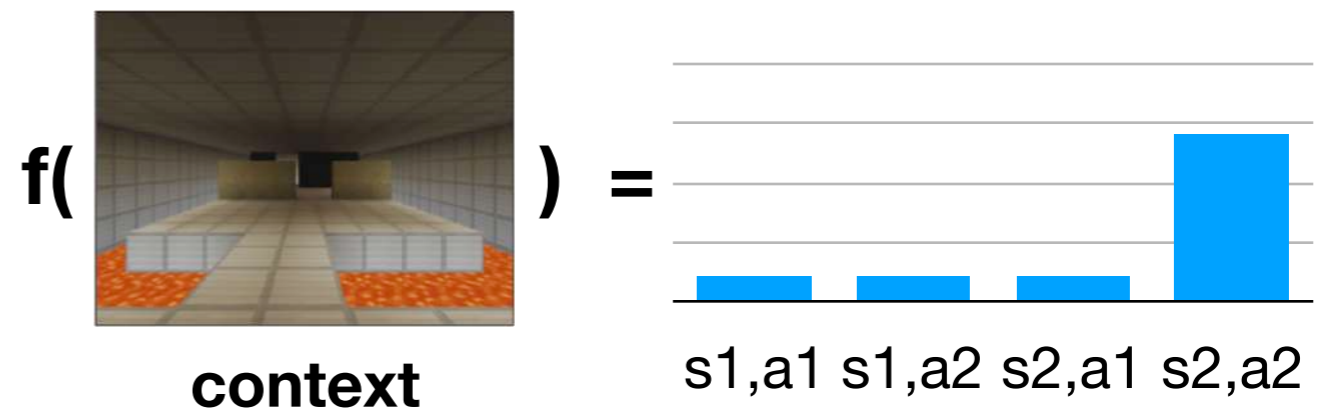


State at level h+1:



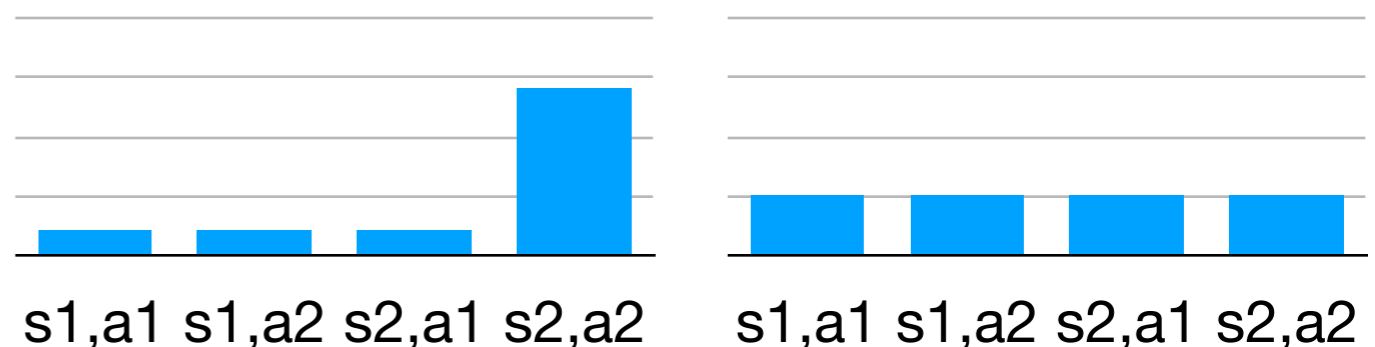
Approach

Our Approach: Learn a function that predicts the conditional probability of (previous state, action) pairs from contexts.
 (assume access a regression oracle to learn this function)

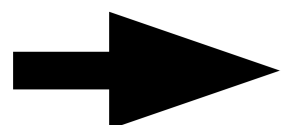


State at level h : $s1, s2$
 Actions: $a1, a2$

Different conditional probabilities correspond to different states



State at level $h+1$:



State classification

Guarantees

Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M, K, H)$ samples in polynomial time, with H calls to supervised learning black box.

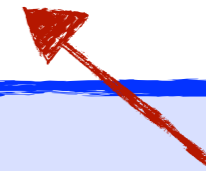
Guarantees

Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M, K, H)$ samples in polynomial time, with H calls to supervised learning black box.

M = Number of hidden states, K = Number of actions, H = Time horizon

Guarantees

Statistical efficiency



Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M,K,H)$ samples in polynomial time, with H calls to supervised learning black box.

M = Number of hidden states, K = Number of actions, H = Time horizon

Guarantees

Statistical efficiency

Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M, K, H)$ samples in polynomial time, with H calls to supervised learning black box.

M = Number of hidden states, K = Number of actions, H = Time horizon

Computational efficiency

Guarantees

Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M, K, H)$ samples in polynomial time, with H calls to supervised learning black box.

M = Number of hidden states, K = Number of actions, H = Time horizon

Statistical efficiency

Computational efficiency

Rich observations

Guarantees

Theorem: Our algorithm can find a near-optimal decoder with $\text{poly}(M, K, H)$ samples in polynomial time, with H calls to supervised learning black box.

M = Number of hidden states, K = Number of actions, H = Time horizon

Statistical efficiency

Computational efficiency

Rich observations

Assumptions

- Supervised learner expressive enough
- Latent states reachable and identifiable

Algorithm details and experiments

@ Poster #208