Instance Based Model Learning

Joseph Xu
University of Michigan
Soar Workshop 2009

Motivation

2 extremes of AI problem solving

Agent simulates domain interactions in its head, plans fully before acting	Agent has no information about domain, does local search
A* search	Watson Q-Learning
Engineer must preprogram all aspects of domain into agent	No do
Agent doesn’t need any experience with the world	Agen the world

Something that makes predictions about the effects of agent’s actions

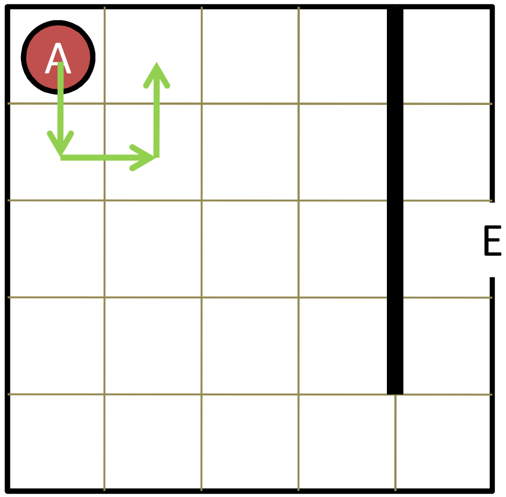
Bridge the gap

- Let the agent learn an (imperfect) domain model from experience with the world
- Agent requires less experience with the world because it generalizes experiences to new situations

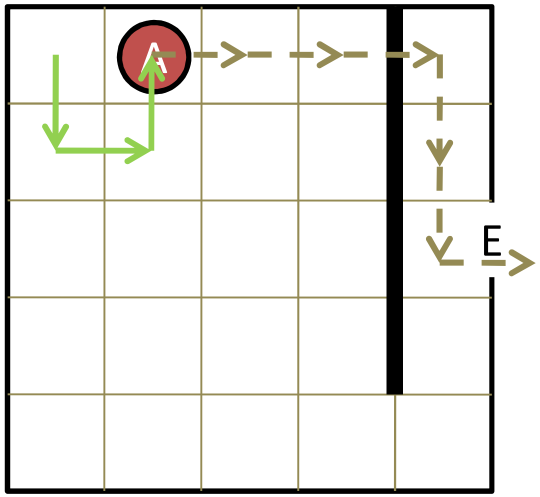
3 Basic Research Questions

1. How to generalize experiences into predictive domain models?
2. How to use potentially imperfect domain models to speed up problem solving?
3. How to do all this in the context of Soar?

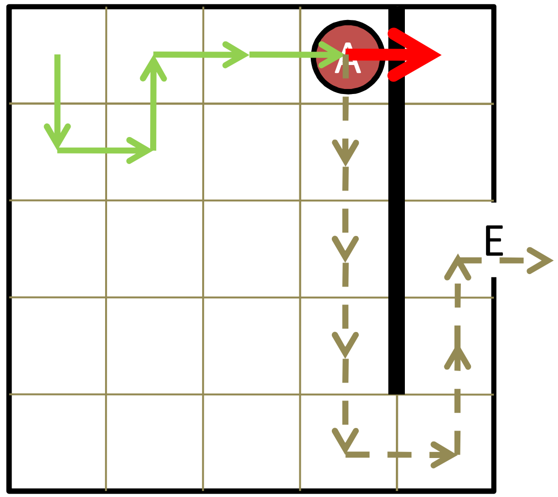
An Example of Integrated Planning, Learning, and Acting



1. Agent initially has no model of the world, so it just wanders

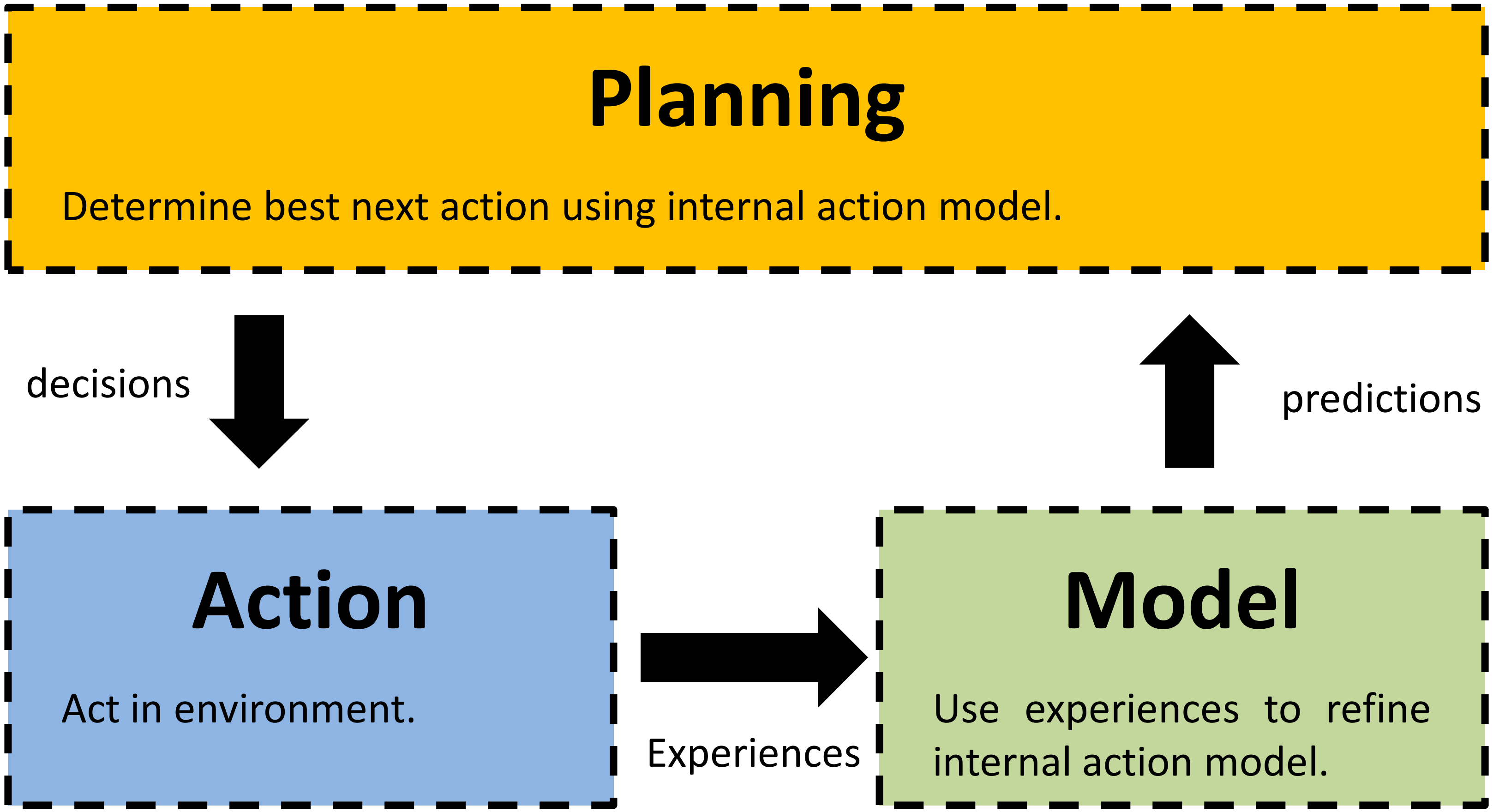


2. Agent learns specific model.
3. Agent generalizes plan, which can lead to over-optimistic path.



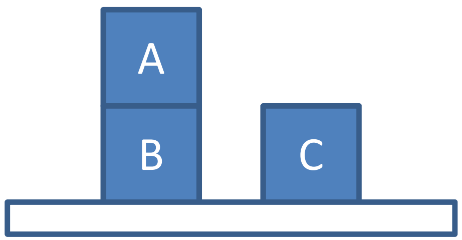
4. Agent detects mismatch between model and world.
5. New experience refines model.
6. Agent replans.

System Overview

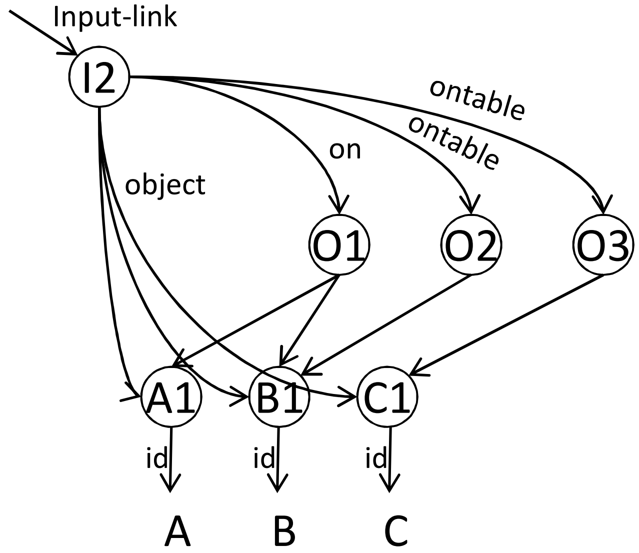


Assumptions About the World

- Deterministic, discrete time steps
- Effects of actions take place in exactly one time step
- Relational representation
  - Only entities are objects, object attributes, and relations on objects
  - Consistent with Soar conventions




```
(object A)
(object B)
(object C)
(on A B)
(ontable B)
(ontable C)
```



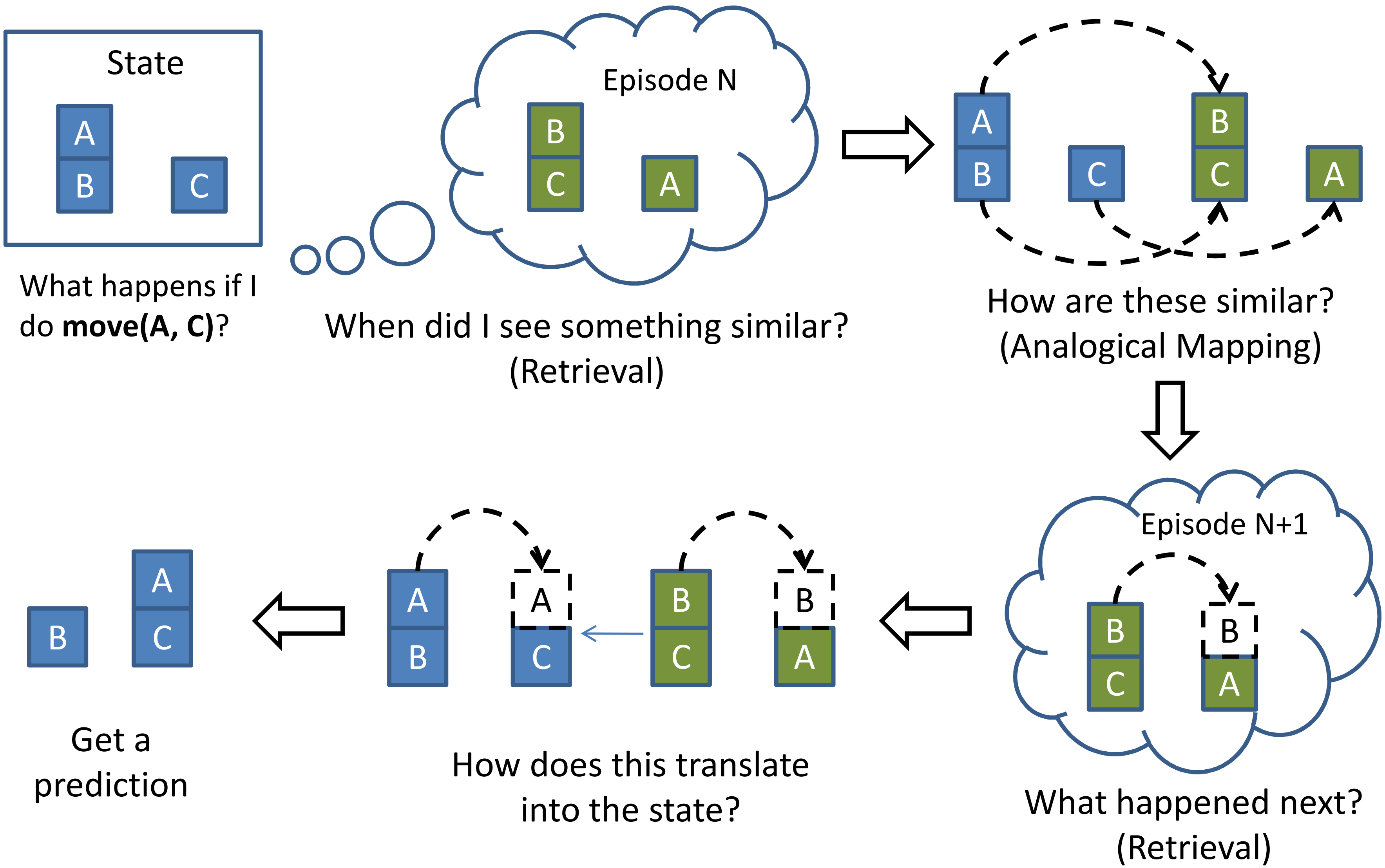
Instance-Based Models

Basic idea

Predict the outcome of an action (state transition) by making an analogy to a previous episode where the action was performed in a similar state

- Needed: memory of the results of previous actions and ability to search for similar past states
  - Episodic memory naturally fits
- The model is the sum total of all previously experienced state transitions
  - Incremental, one-shot learning
  - More experiences means closer analogies, more likely to be correct
  - Will always converge to perfect accuracy

Episodic Memory Based Models



Learning Good Cues

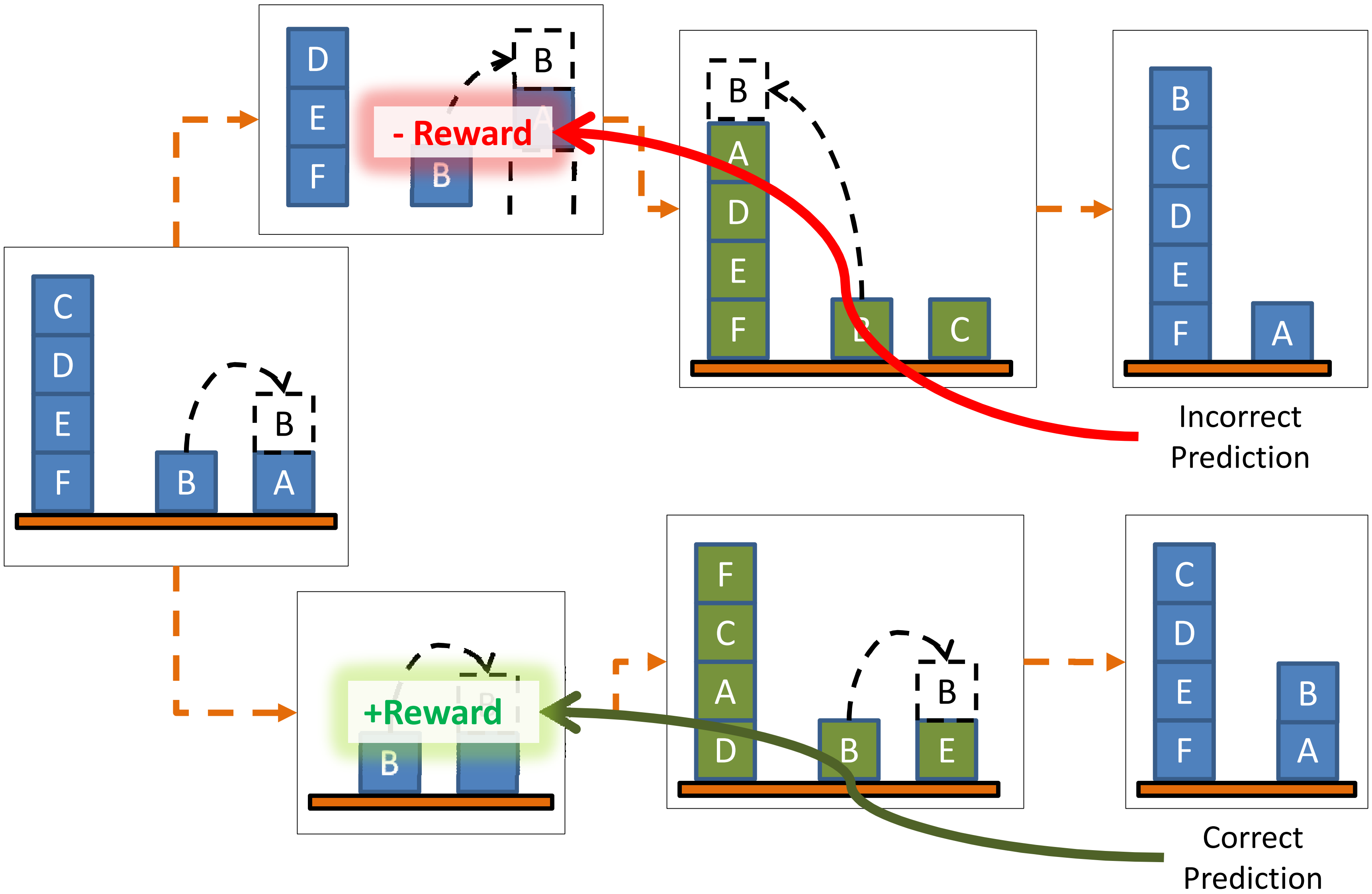
- Epmem will try to match cue as much as possible
- Naïve approach is to use entire current state as cue
- State will contain many features that don’t play a part in determining action effects
- If these distracters are included in the cue, the retrieved state might not be similar in terms of the relevant features

Answer:

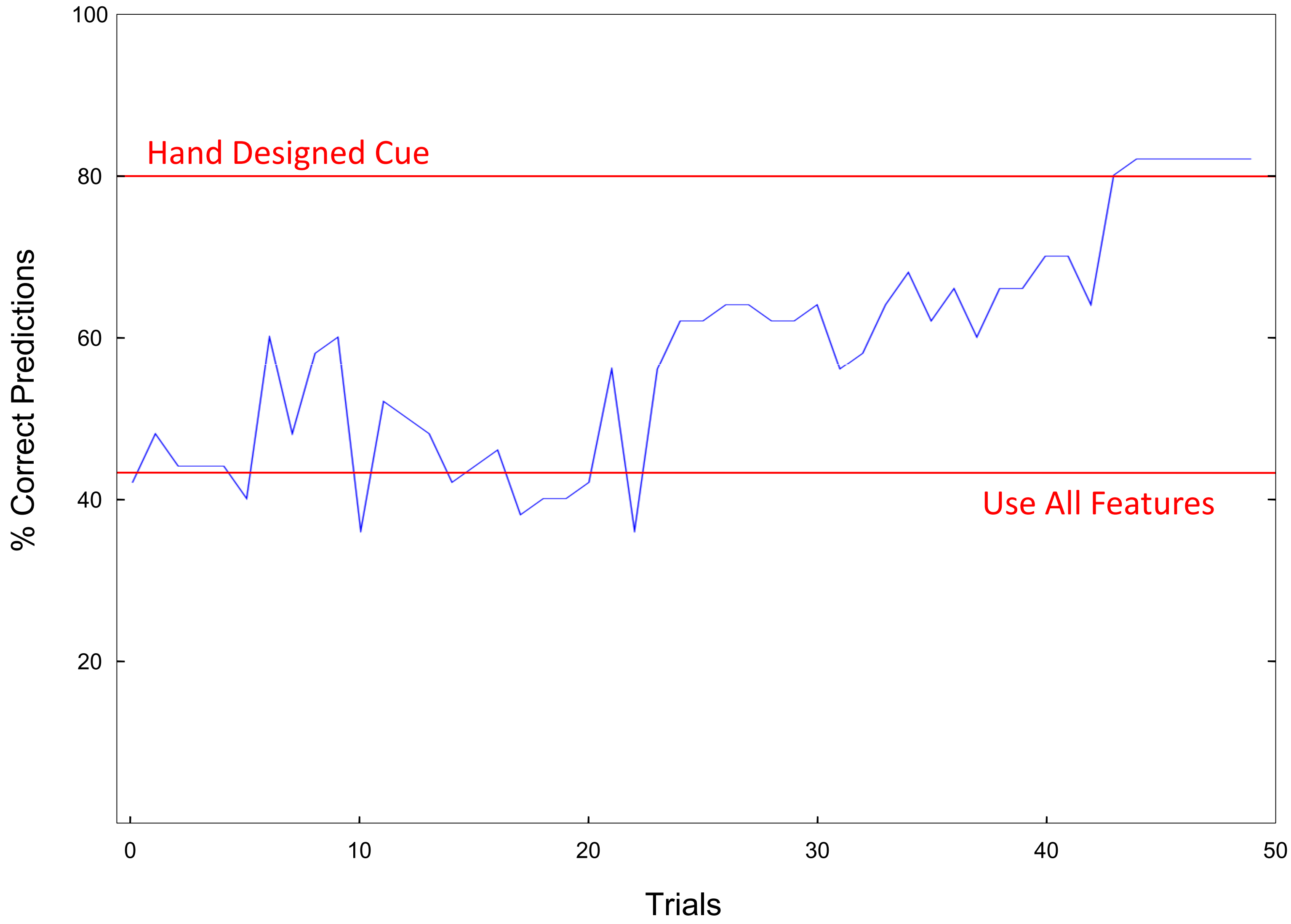
Learn to exclude distracters from cue with reinforcement learning

Thanks Nick

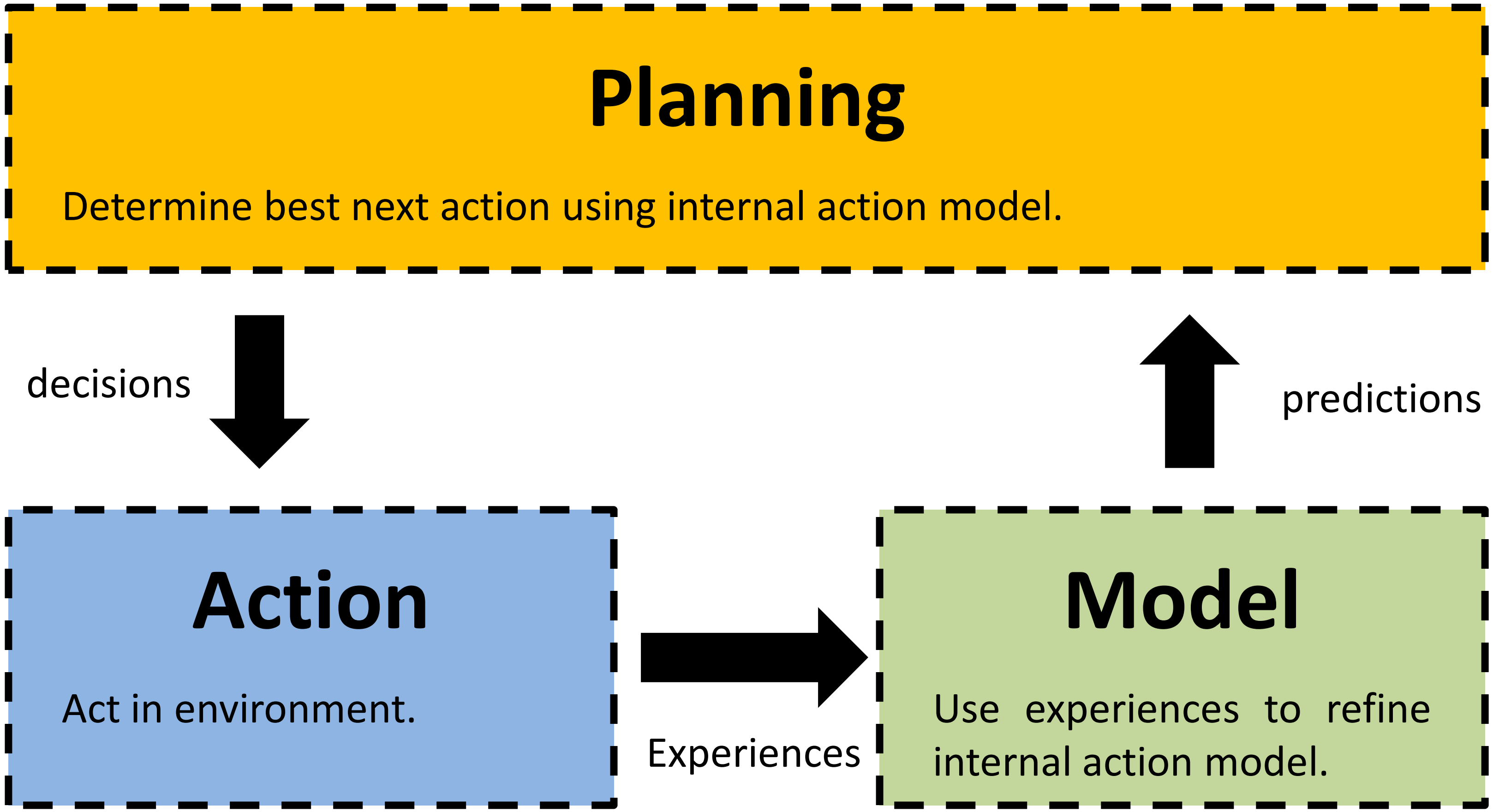
Learning Relevant State Features



Learning Relevant State Features



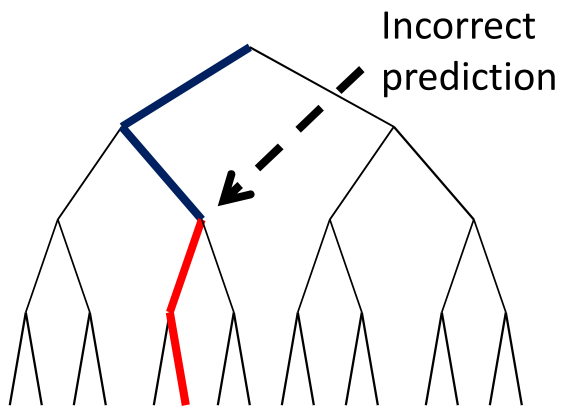
System Overview



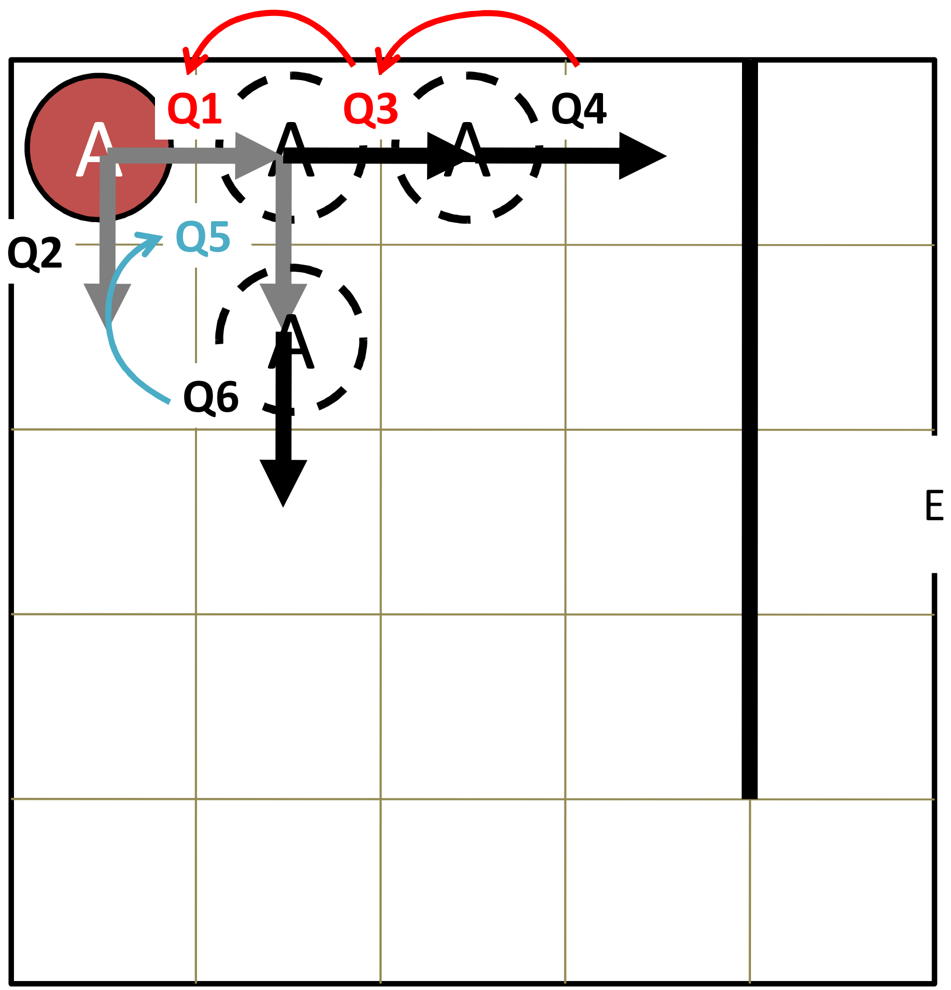
Planning with Learned Models

How do we use possibly imperfect models?

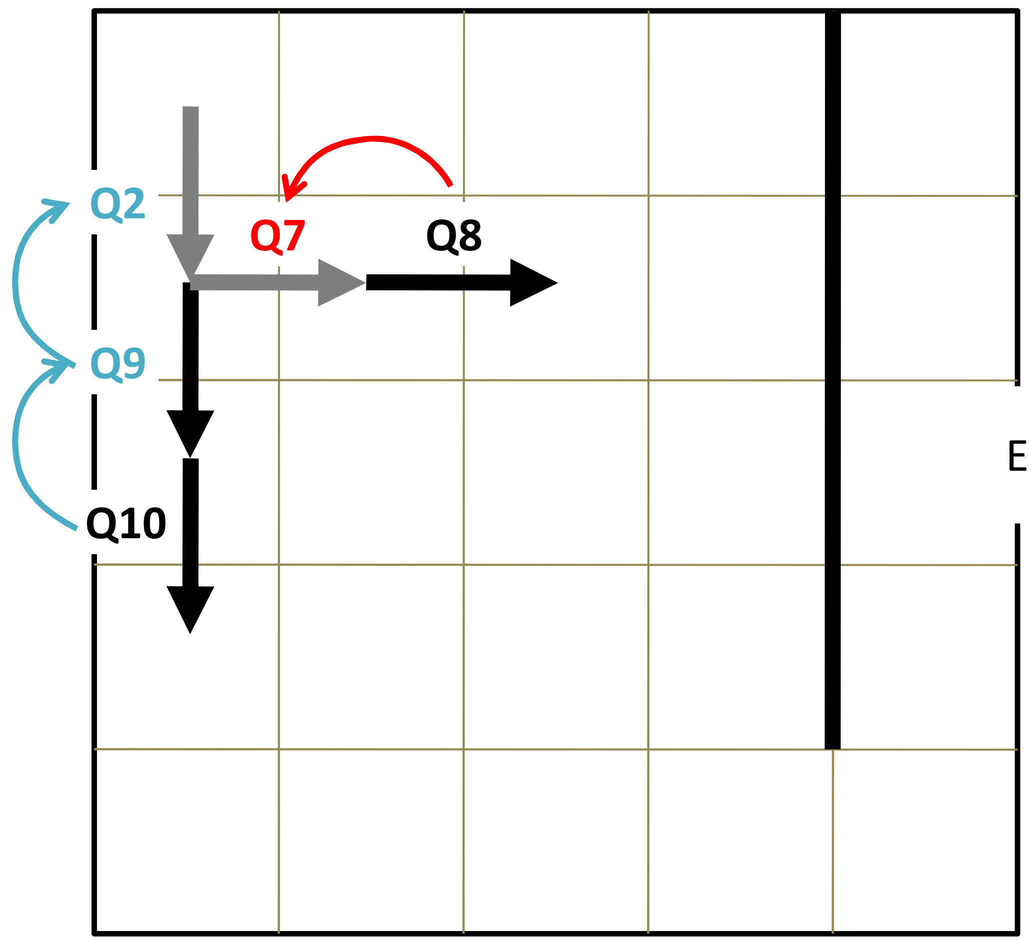
- Problem space search
  - Open-loop policies are vulnerable to single wrong predictions
  - Partial look-ahead is worthless
- Combine look-ahead search with RL
  - Regular Q-learning
  - Use model for shallow look-aheads
  - Back up Q-values
  - Closed loop policies are robust to wrong predictions



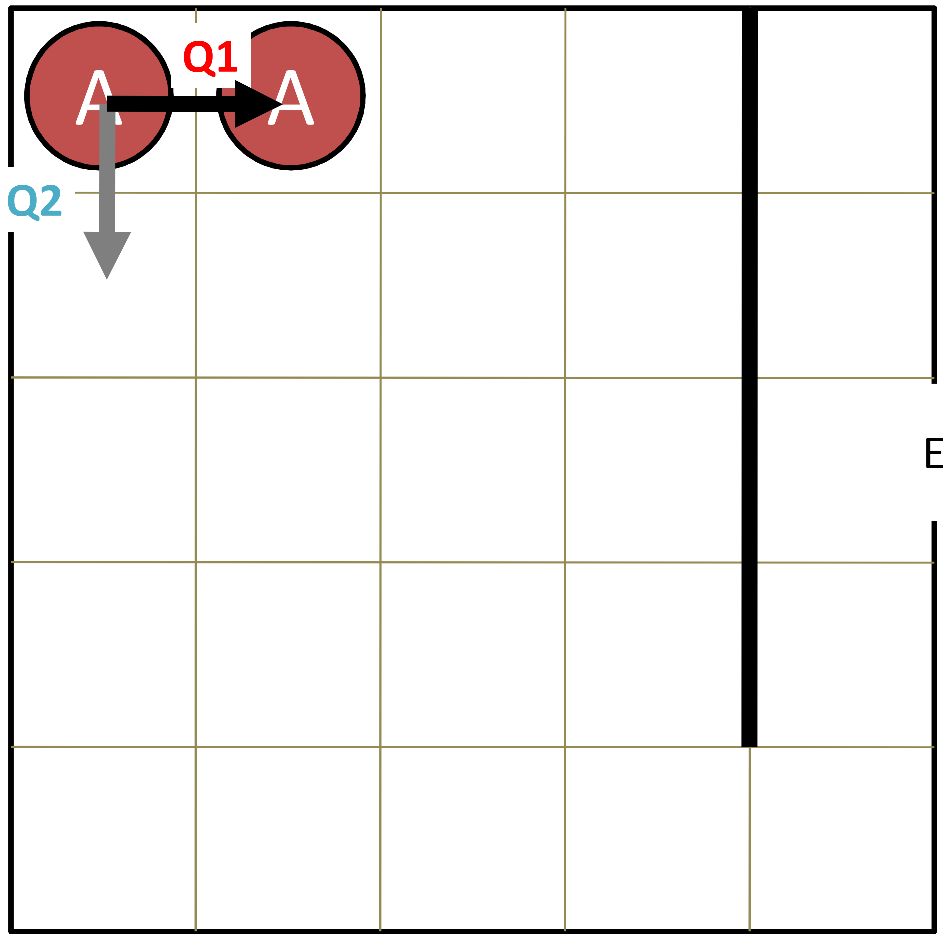
RL with 2 Step Look-ahead



RL with 2 Step Look-ahead



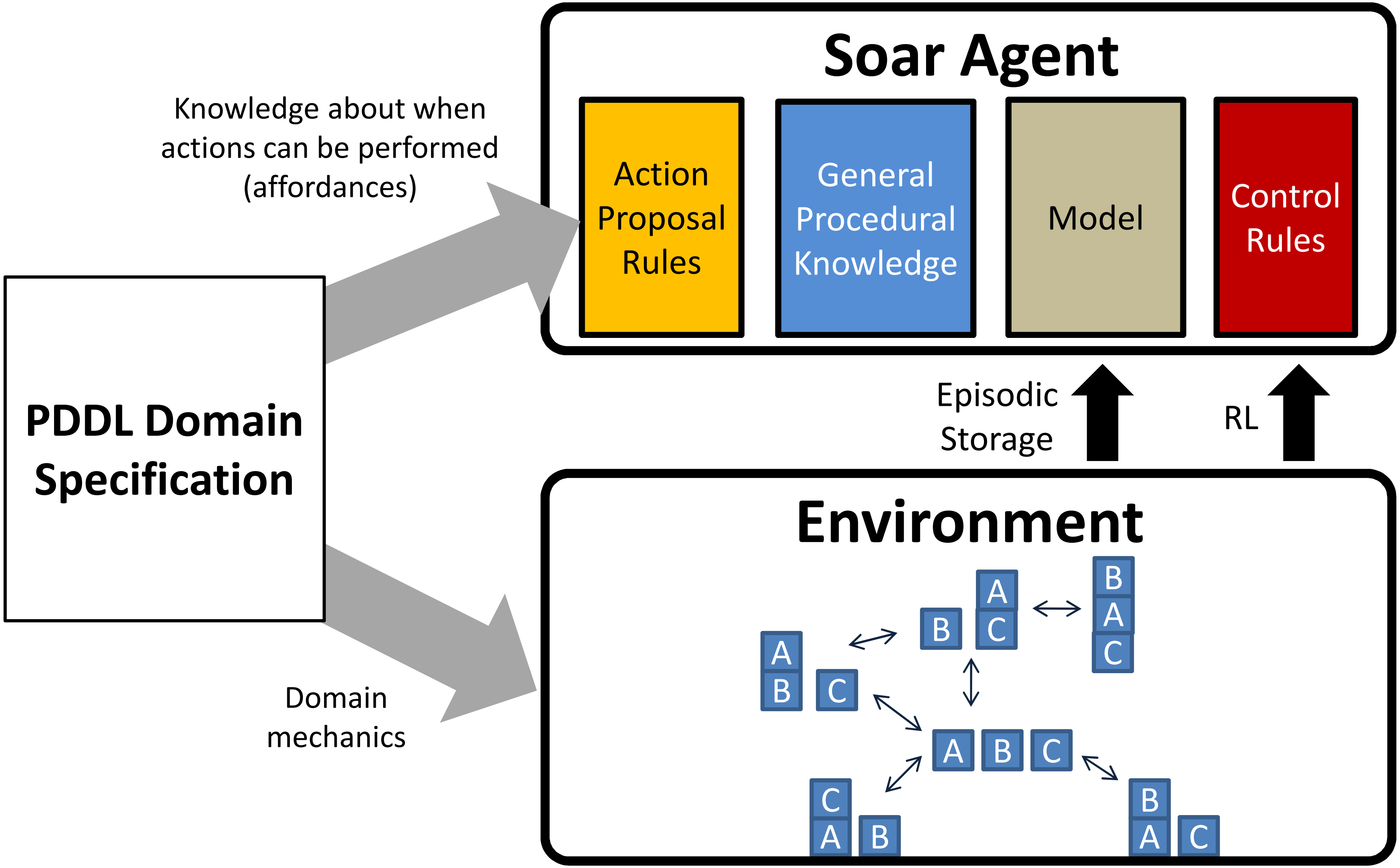
RL with 2 Step Look-ahead



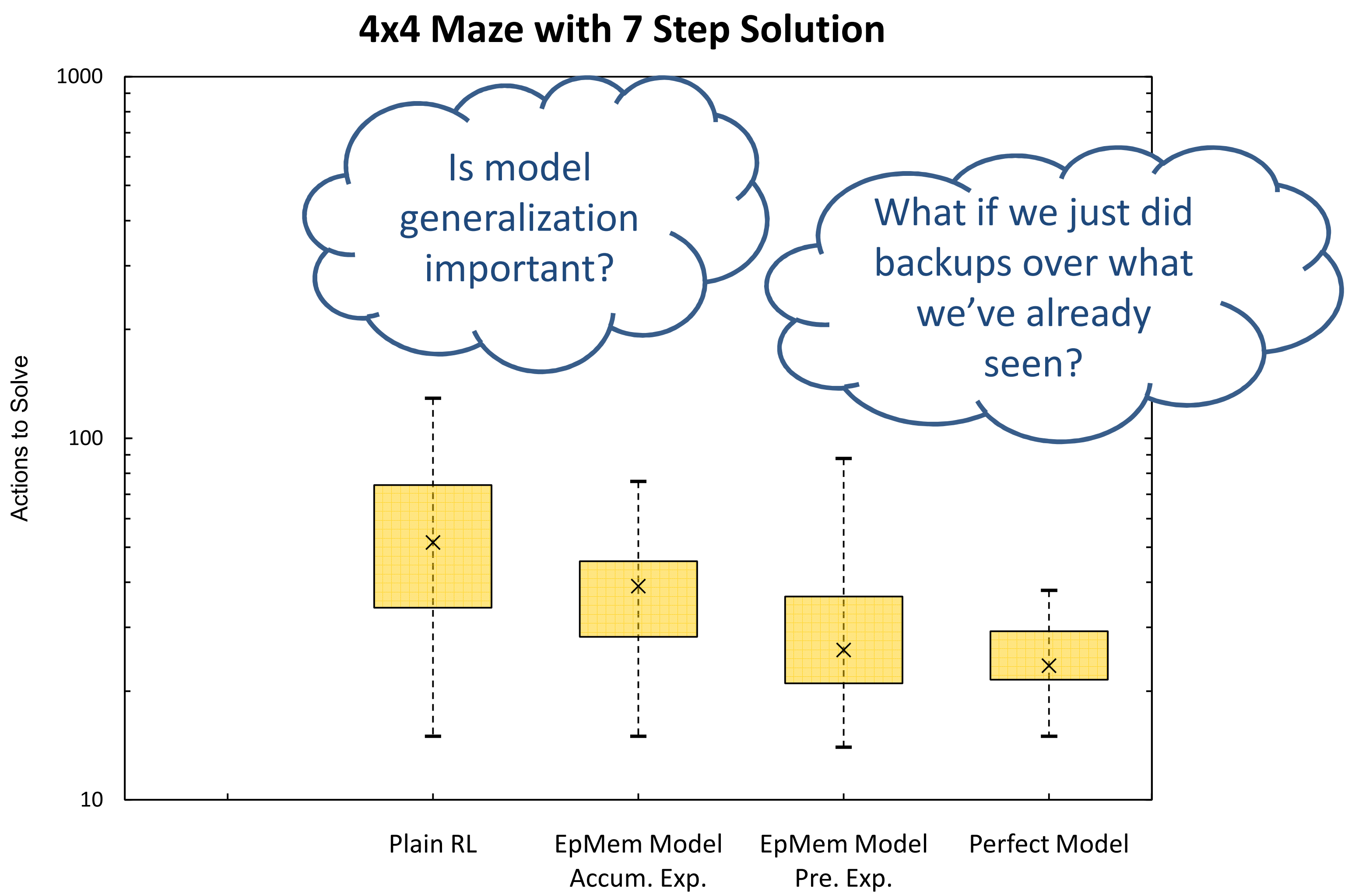
Considerations

- How do we trade off number of real actions and imaginary updates?
  - Look-ahead branching factor, depth
- Bad models lead to bad back-ups
  - Agent should hold off on look-ahead until it has some confidence in model accuracy
  - How to define confidence in model accuracy?

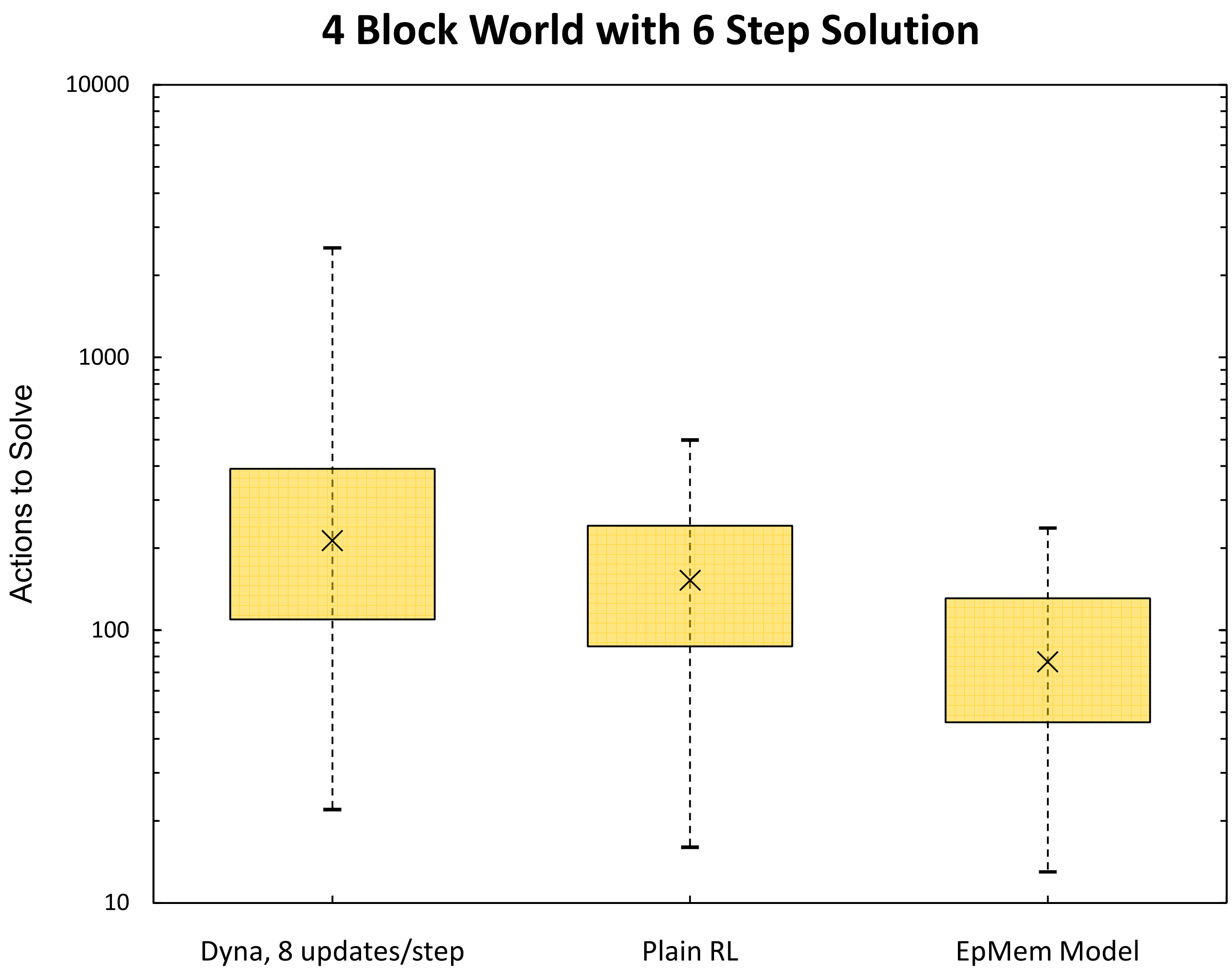
Experimental Setup



Preliminary Results



Preliminary Results



Future Work

- Chunk over episodic retrievals to get procedural domain knowledge
- Consider other sources of knowledge when doing prediction
  - Domain independent semantic knowledge such as naïve physics models, object category information
    - YJ’s work – learning semantic categories
- Harder domains – Rogue?

Golden Nuggets

- Model learning is incremental
- Models are guaranteed to converge to perfection
- Can handle any relational domain

Chunks of Coal

- Many algorithms are slow
- Analogical mapping algorithm is naïve
- Results are from trivial problems