A Framework for Hierarchical, Probabilistic Planning and Learning

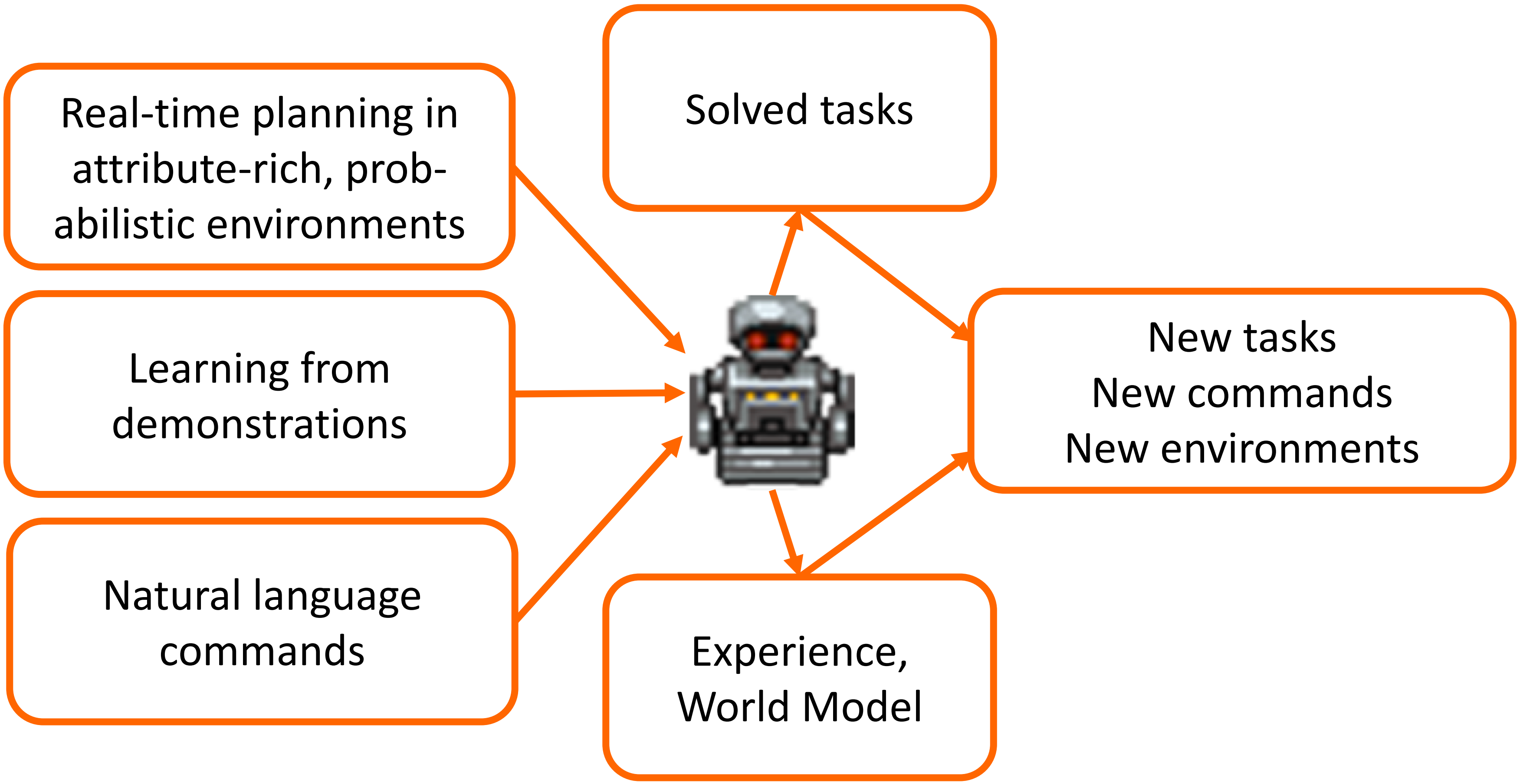
Stefanie Tellex
Marie desJardins
Michael Littman
Cynthia Matuszek

UMBC
AN HONORS UNIVERSITY IN MARYLAND

BROWN

Planning in Robotics Domains

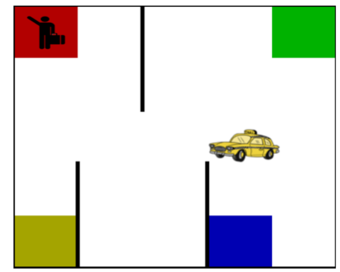
Varied, huge, stochastic, and messy

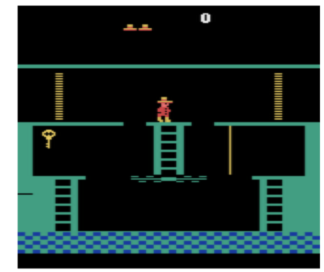


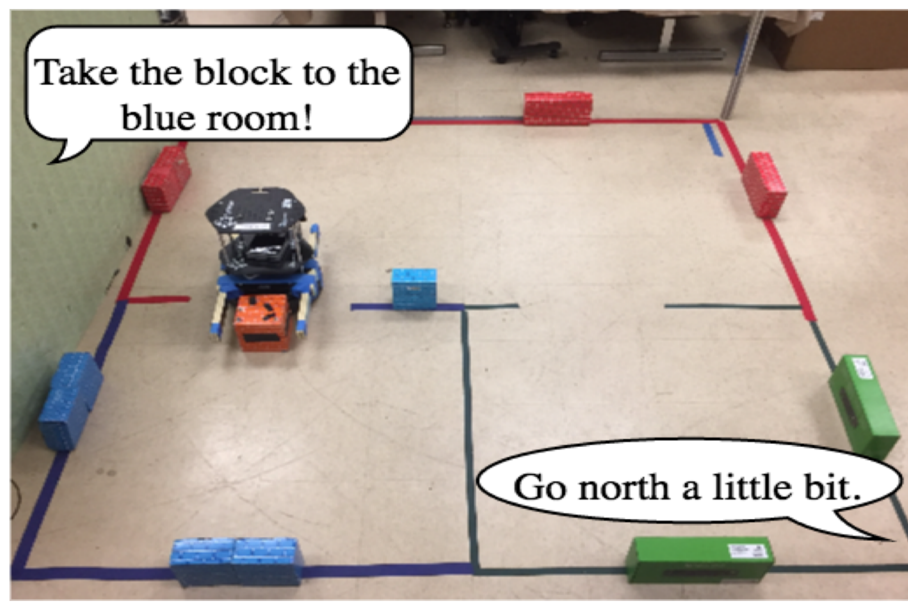
Objectives

Efficient, non-domain-specific planning

1. Learn hierarchies from data
2. Plan efficiently within them
3. Learn language groundings to drive tasking

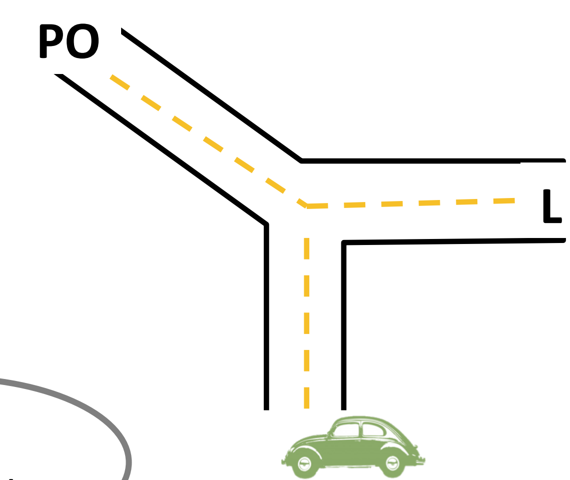






Non-Hierarchical Planning

- Without hierarchy, consider all possible action sequences from all states



Go to Post Office

Go to Library

Post office:

- Turn L 60˚
- Turn R 90˚
- Straight
- Turn R 90˚

There yet?

Turn L 60˚

Turn L 90˚

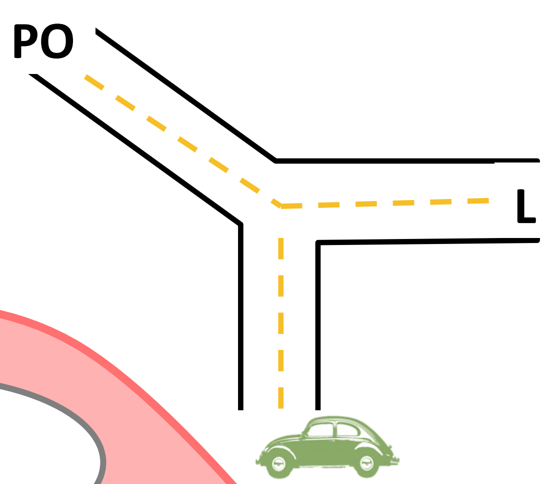
Turn R 60˚

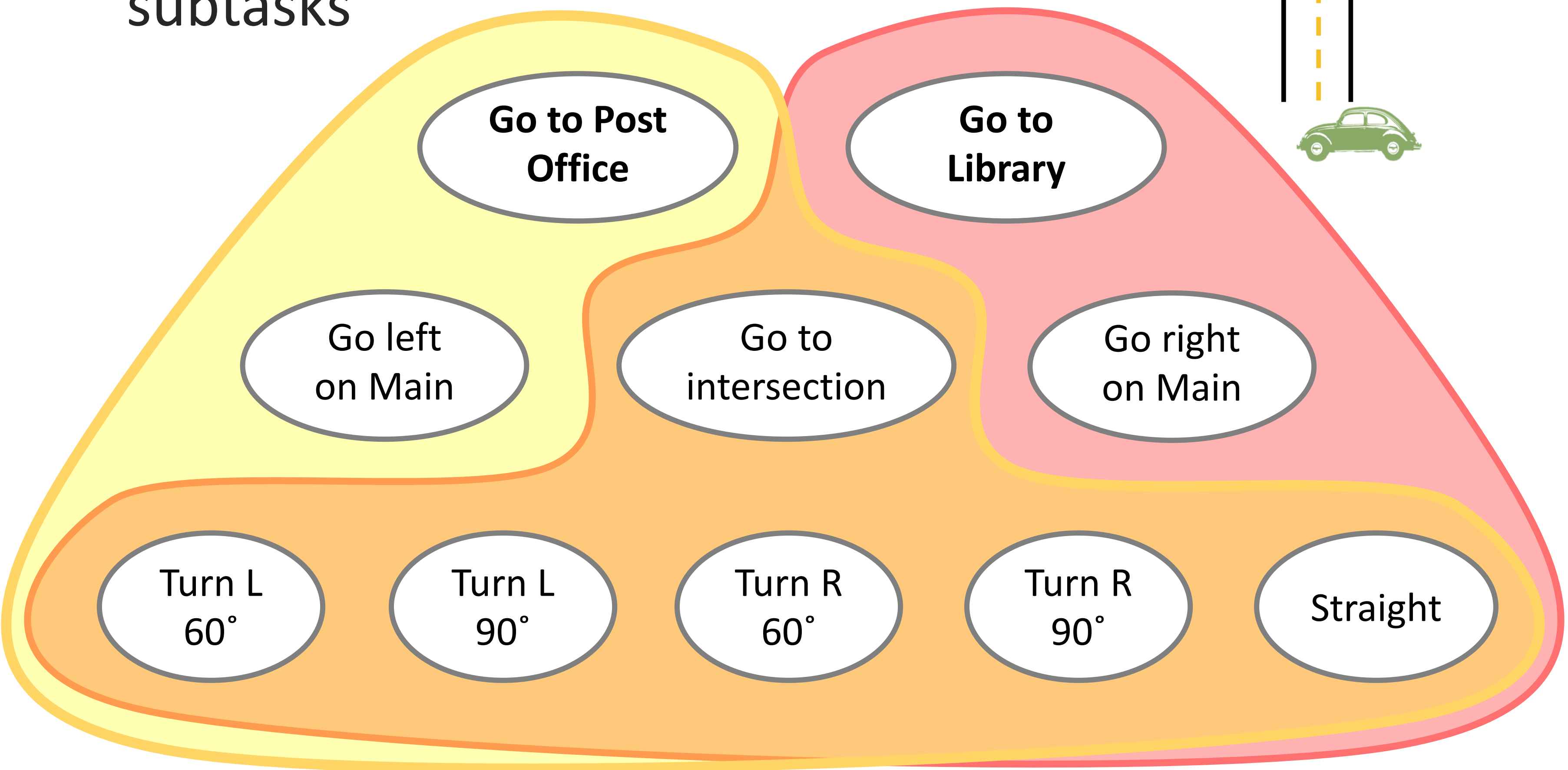
Turn R 90˚

Straight

Top-Down Hierarchical Planning

- Given hierarchy, consider possible sequences involving relevant subtasks

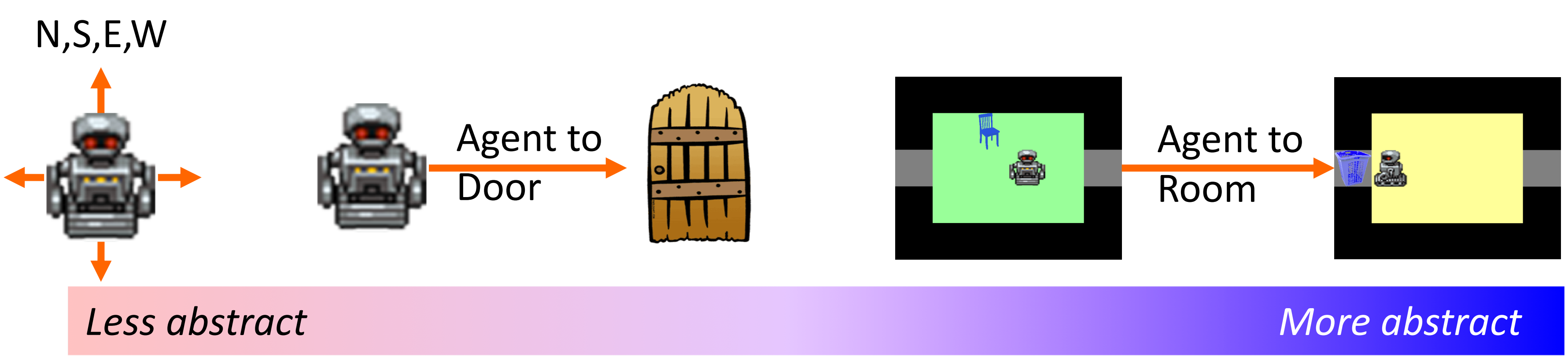




Abstract MDPs (AMDPs)

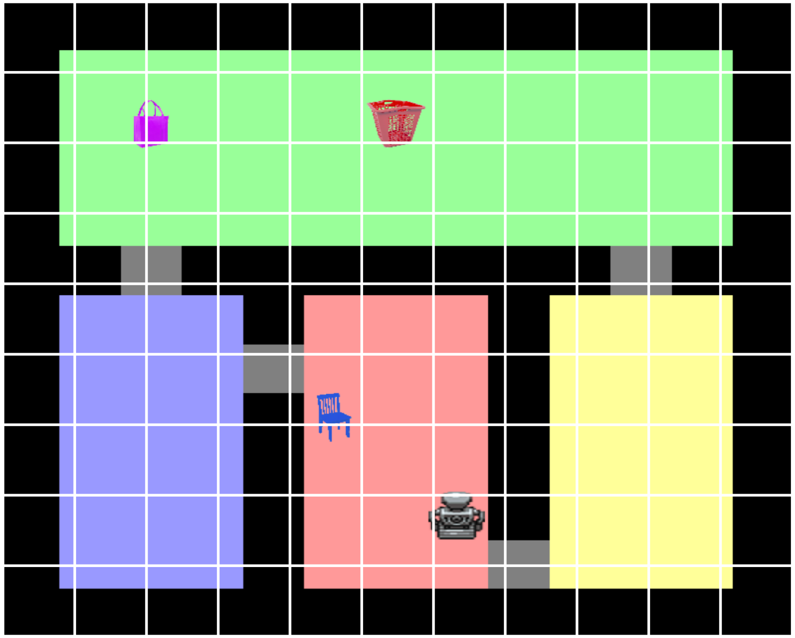
Markov Decision Processes, plus abstraction:

- MDP (MDP): $<\mathcal{S}, \mathcal{A}, T, R, \varepsilon>$:
  - States, actions, transitions, rewards, terminal states
- Abstract MDPs add **state mapping functions**: $<\mathcal{S}, \mathcal{A}, T, R, \varepsilon, F>$
- $F^{\ell} : s \longrightarrow s^{\ell}$ projects states from ground level to current level of abstraction



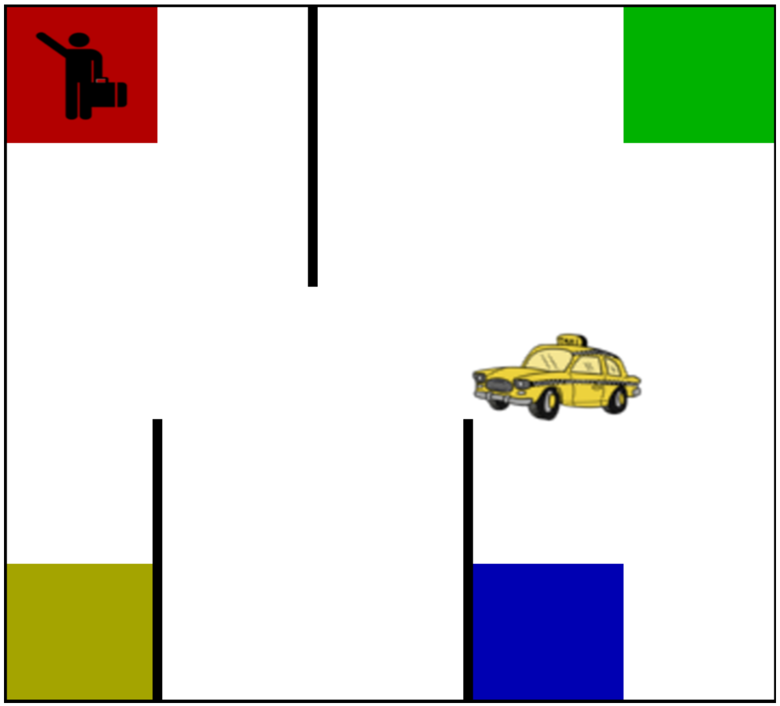
Projecting to Abstract States

- Cleanup task requires going red → green → blue
- Don’t need exact location when planning next room to visit
- X, Y coordinates **projected up to** appropriate level

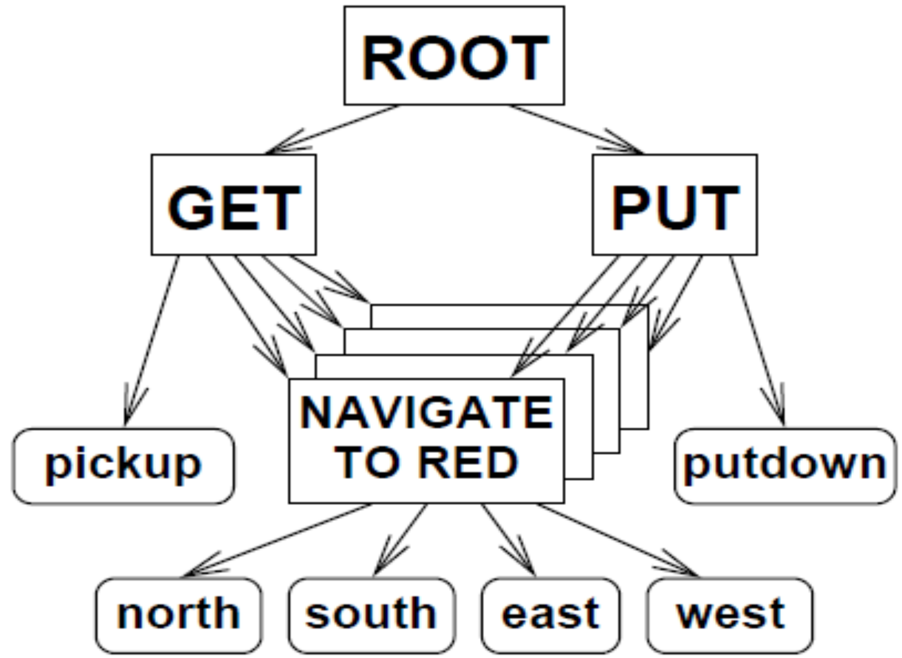


- loc<7,2> $\xrightarrow{F^1}$ loc<“red room”>

Classic Taxi Domain + Task Hierarchy



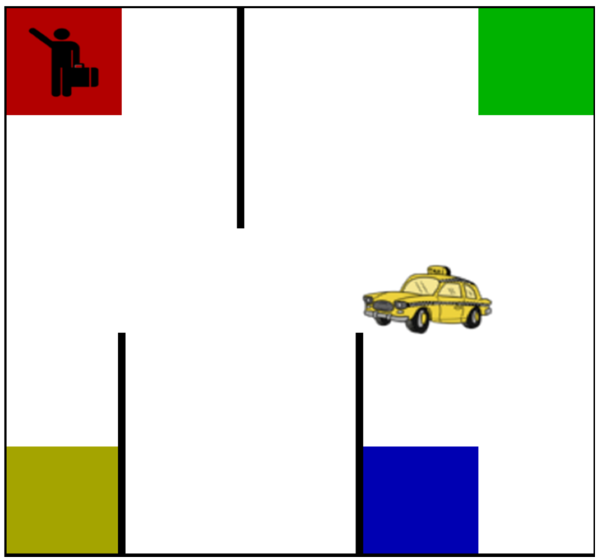
Taxi Domain (Dietterich, 2000)
Agent is taxi, must take passenger to depot (red, yellow, green, blue)

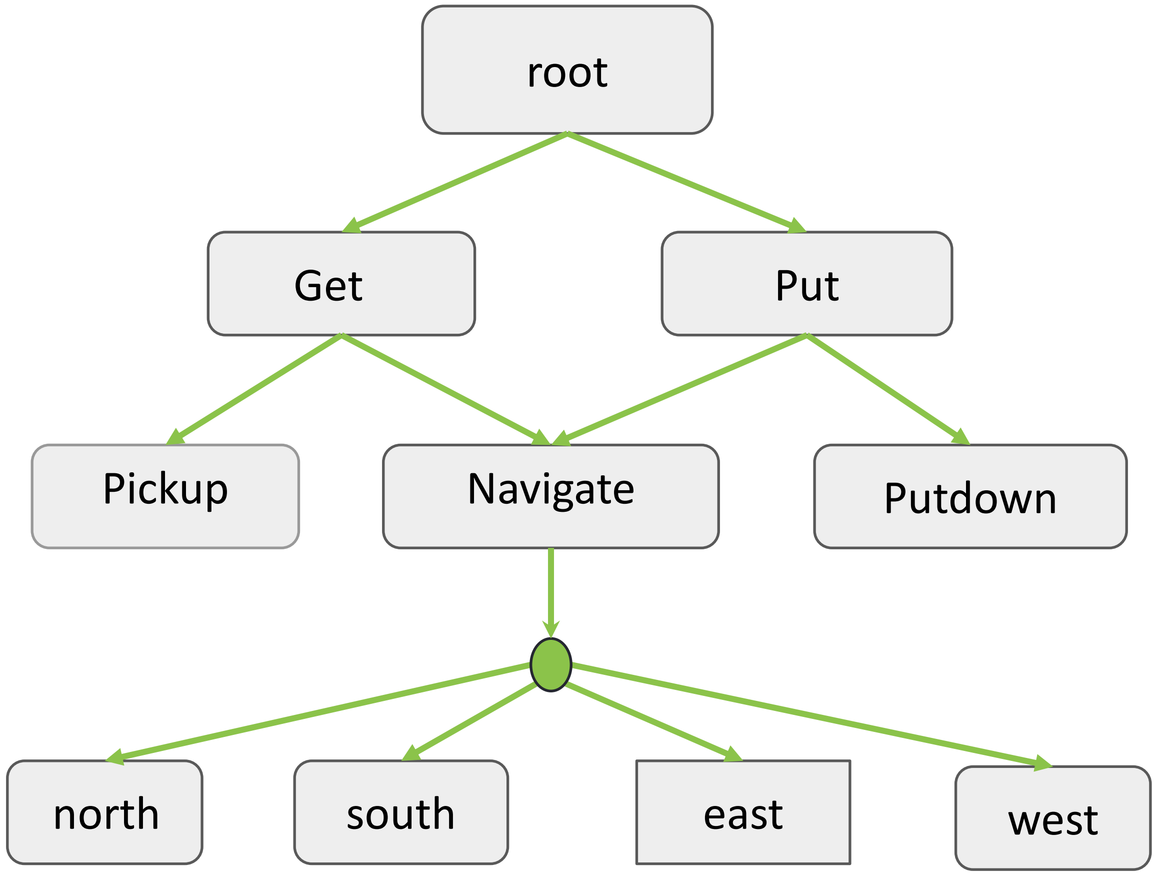


A Task Hierarchy for Taxi
(rectangles are subgoals,
leaf nodes are ground actions)

[Dietterich 2000]

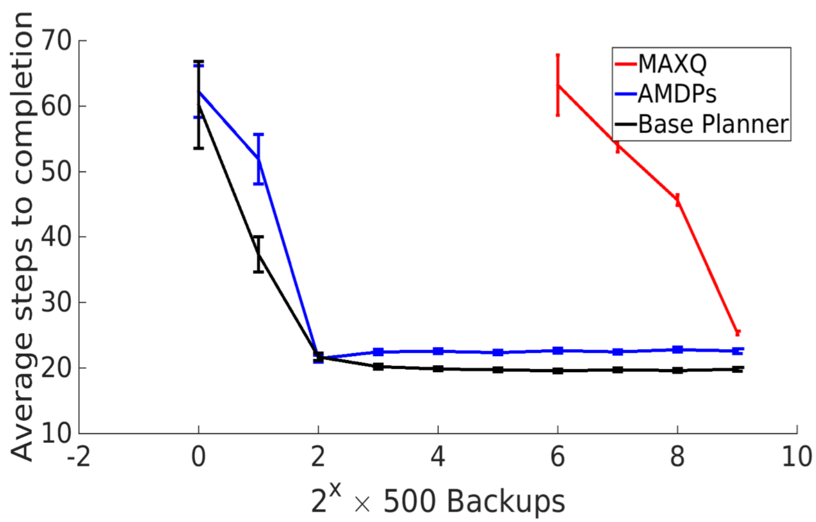
Taxi Representation as AMDP



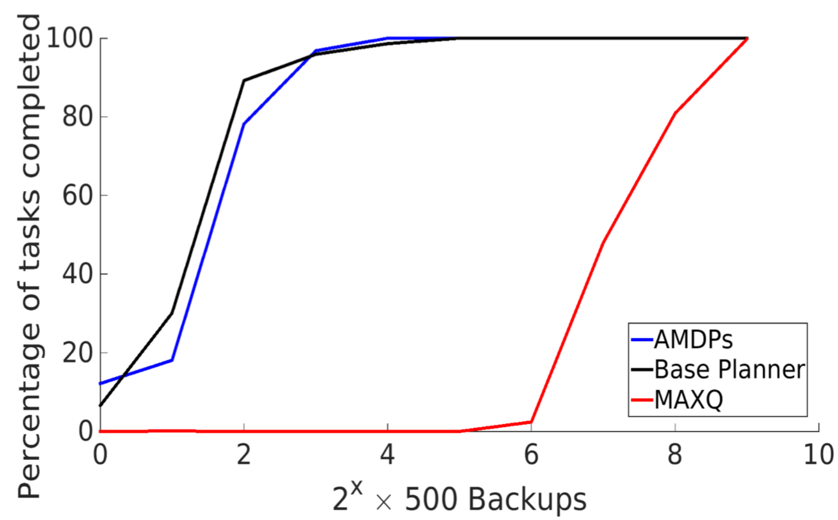


AMDP Planning: Taxi

- Domain is too small to benefit

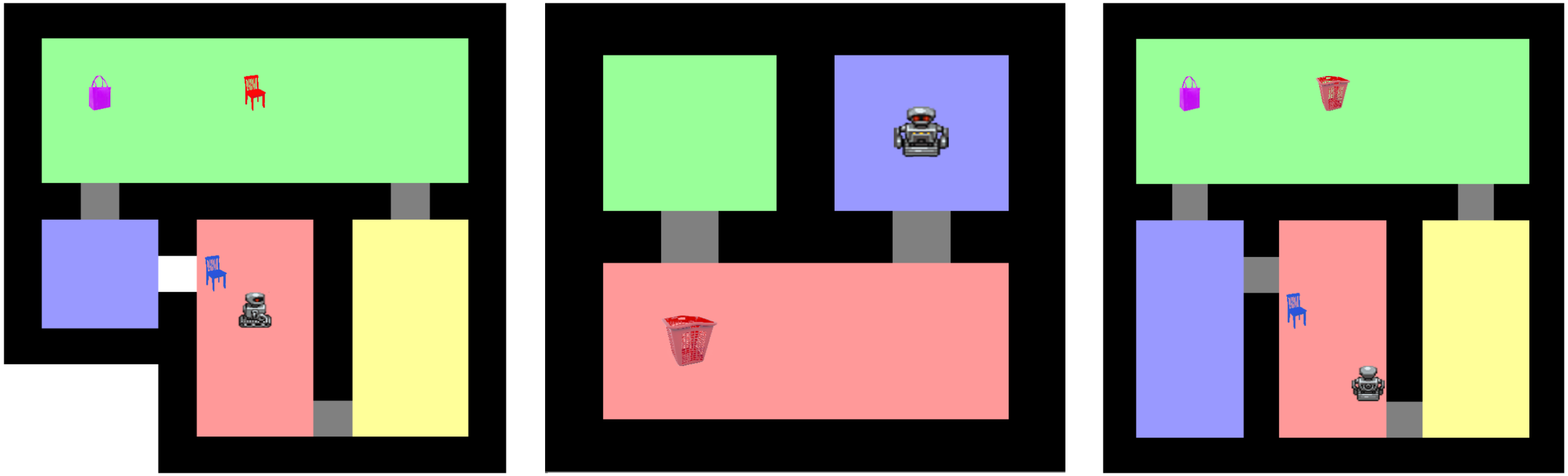


Lower is better

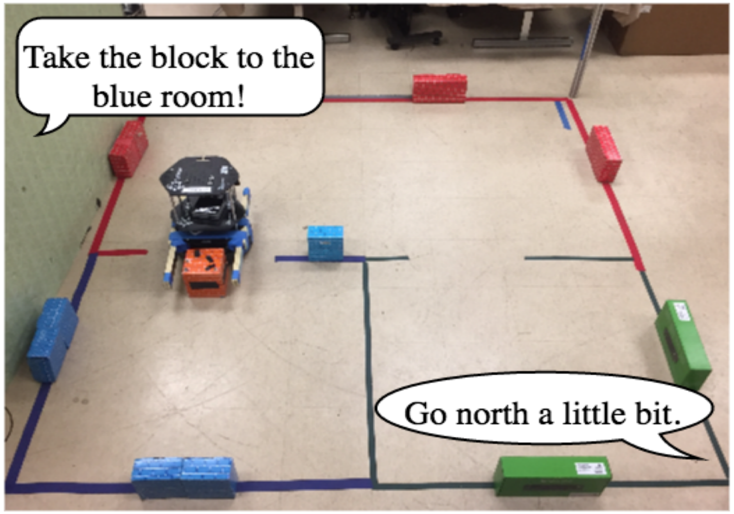


Higher is better

Cleanup Domain

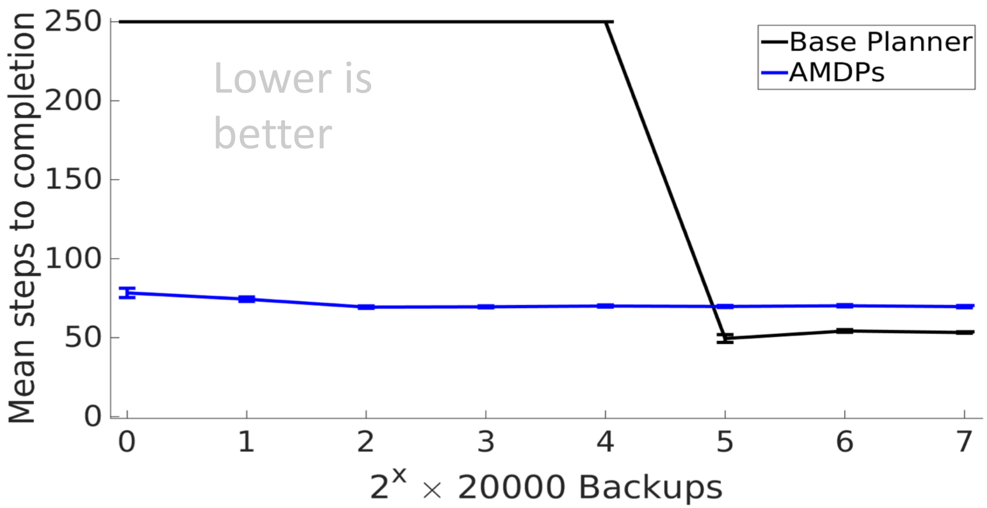


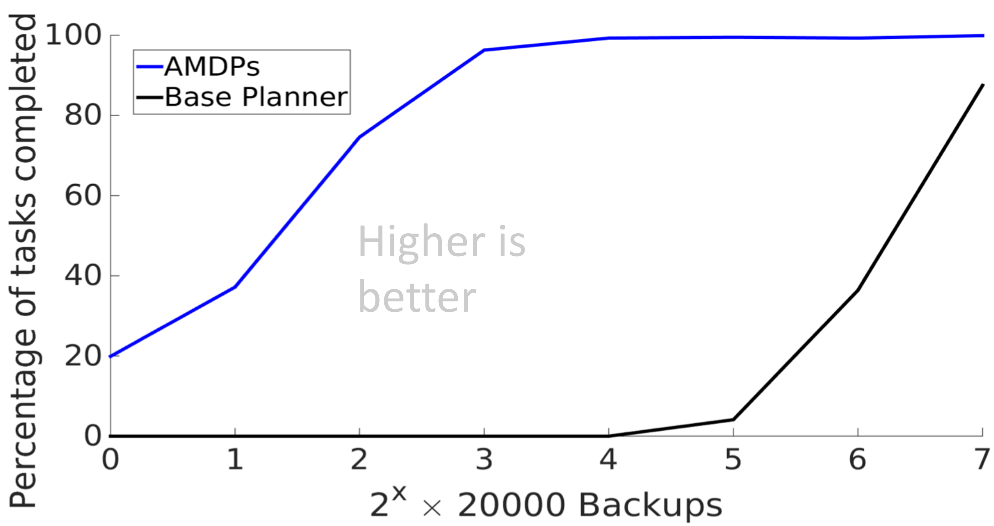
- **State:** Agent and object location / orientation, door lock boolean
- **Actions:** N, S, E, W, Pull
  - Stochastic transitions possible.
- **Objectives:** Take specified object or agent to specified room



Planning over AMDPs in Cleanup

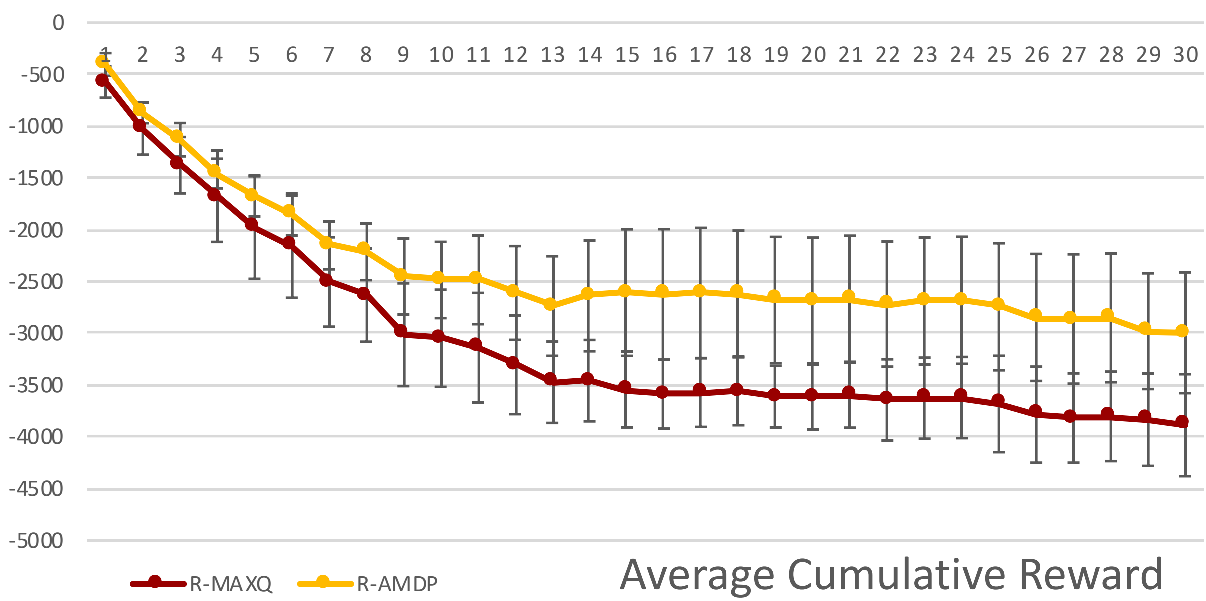
- Complex task
  - Many objects
  - Highly combinatorial
- AMDPs **start** finding solutions much faster
  - Fewer backups compared to optimal solver





R-MAX + AMDPs (R-AMDPs) / PALM

- Plan top-down starting at R-AMDP-Plan(*H*, *Root*)
  - Determine next action
  - Ground to subgoal (A)MDP
  - Recurse to ground MDP
  - On return, update model for T and R

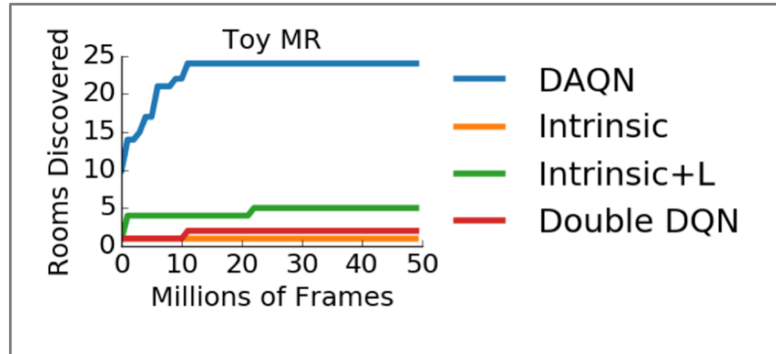


2. Learning Hierarchies

Strategies:

- Quality of behaviors derived from types of approximate abstractions (ICML 2016)
- Combine deep reinforcement learning with model-based approaches using expert-provided state abstractions (AAMAS 2018)
- Learn AMDP hierarchies, rewards, and transition functions directly from data (AAAI 2019, *under submission*)

None of these is perfect!



$$\eta_{model} = \frac{1 + \gamma\left(|\mathcal{S}_G| - 1\right)}{(1-\gamma)^3}$$

$$\eta_{bolt} = \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + \varepsilon k_{bolt} + k_{bolt}\right)}{(1-\gamma)^2}$$

$$\eta_{mult} = \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + k_{mult}\right)}{(1-\gamma)^2}$$

Approximate State Abstractions

- **Approximate state abstractions**: nearly-identical situations ≡ equivalent

$$\forall_{s \in \mathcal{S}_G} V_G^{\pi_G^*}(s) - V_G^{\pi_{GA}}(s) \leq 2\varepsilon\eta_f$$

- Q functions
- Transition and Reward Function
- Boltzmann Distributions over agent actions

$$\eta_{Q^*} = \frac{1}{(1-\gamma)^2}$$

$$\eta_{model} = \frac{1 + \gamma\left(|\mathcal{S}_G| - 1\right)}{(1-\gamma)^3}$$

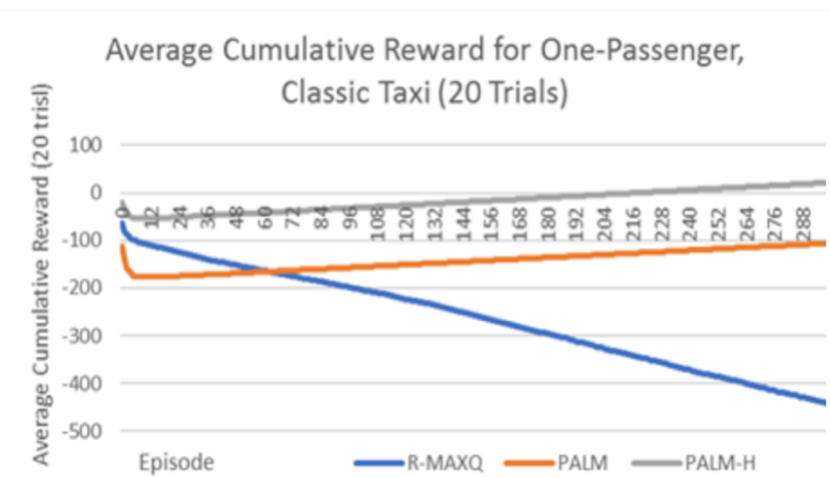
$$\eta_{bolt} = \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + \varepsilon k_{bolt} + k_{bolt}\right)}{(1-\gamma)^2}$$

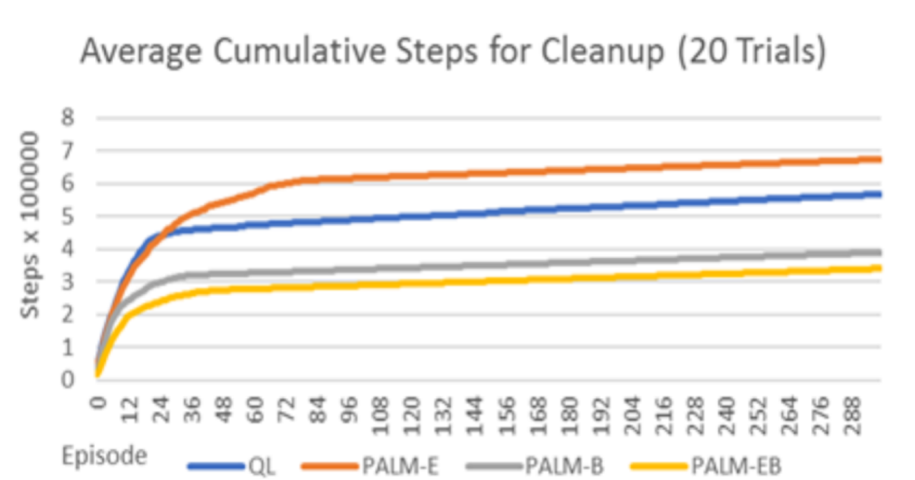
$$\eta_{mult} = \frac{\left(\frac{|\mathcal{A}|}{1-\gamma} + k_{mult}\right)}{(1-\gamma)^2}$$

[Dave Abel, D. Ellis Hershkowitz & Michael L. Littman]

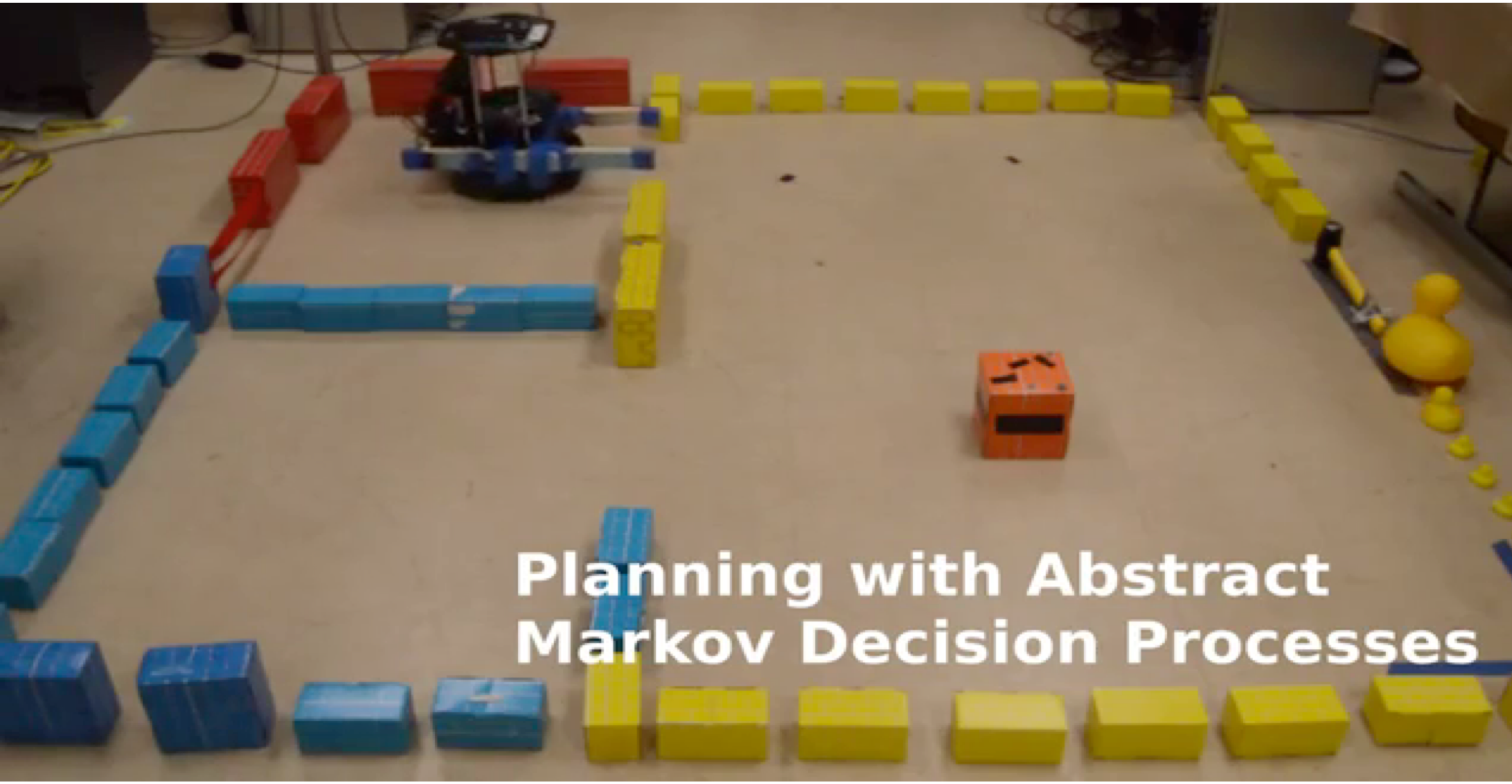
Deep Abstract Q Networks

- Model learning with R-Max to learn AMDP transition and reward models.
- HierGen to learn hierarchies for tasks using data provided by example solution trajectories





Planning Example

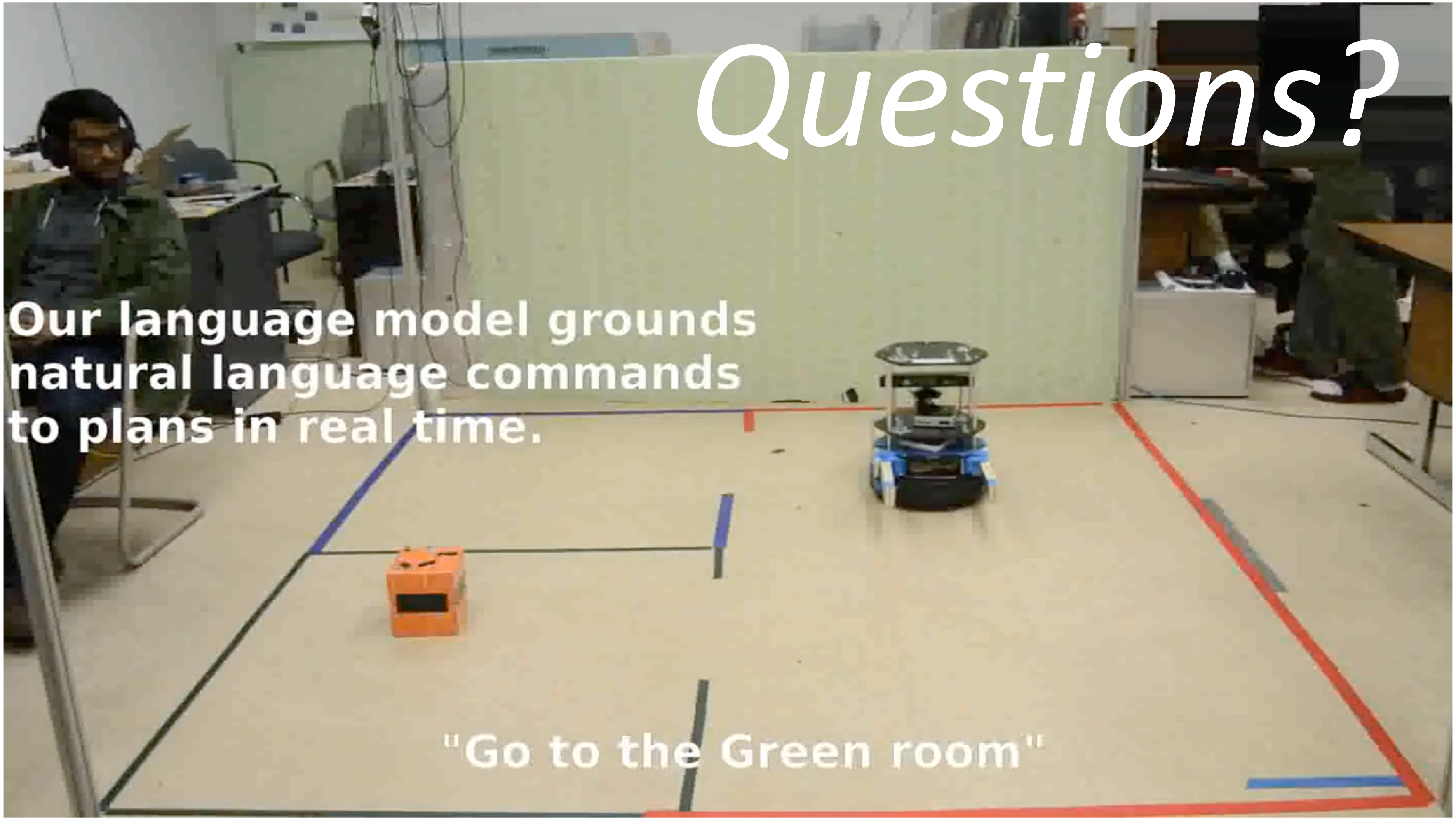


[Gopalan et al. ICAPS-17]

Publications

- Near Optimal Behavior via Approximate State Abstraction - David Abel, D. Ellis Hershkowitz, Michael L. Littman. ICML 2016
- Planning with Abstract Markov Decision Processes - Nakul Gopalan, Marie desJardins, Michael L. Littman, James MacGlashan, Shawn Squire, Stefanie Tellex, John Winder, Lawson L.S. Wong. Abstraction in Reinforcement Learning Workshop @ ICML 2016.
- Planning with Abstract Markov Decision Processes - Nakul Gopalan, Marie desJardins, Michael L. Littman, James MacGlashan, Shawn Squire, Stefanie Tellex, John Winder, Lawson L.S. Wong. ICAPS 2017.
- Deep Abstract Q-Networks - Melrose Roderick, Christopher Grimm, Stefanie Tellex. Hierarchical RL workshop @ NIPS 2017.
- RAMDP: Model-Based Learning for Abstract Markov Decision Process Hierarchies - Shawn Squire, John Winder, Matthew Landen, Stephanie Milani, and Marie desJardins). Third Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM).
- Towards Planning With Hierarchies of Learned Markov Decision Processes - John Winder, Shawn Squire, Matthew Landen, Stephanie Milani, and Marie desJardins. ICAPS workshop on Integrated Execution (IntEx) 2017.
- Deep Abstract Q-Networks - Melrose Roderick, Christopher Grimm, Stefanie Tellex. AAMAS 2018.

Future Work


Questions?
Our language model grounds natural language commands to plans in real time.
"Go to the Green room"