Toward Affordance-Aware Planning

David Abel & Gabriel Barth-Maron
James MacGlashan, Stefanie Tellex

Dept. of Computer Science, Brown University

RSS: First Workshop on Affordances
July 13, 2014

Robotics Motivation

Tool Use, Subgoal Planning

[1] Jennifer L. Barry https://www.youtube.com/watch?v=7f6jFpNcNV4

Problem Statement

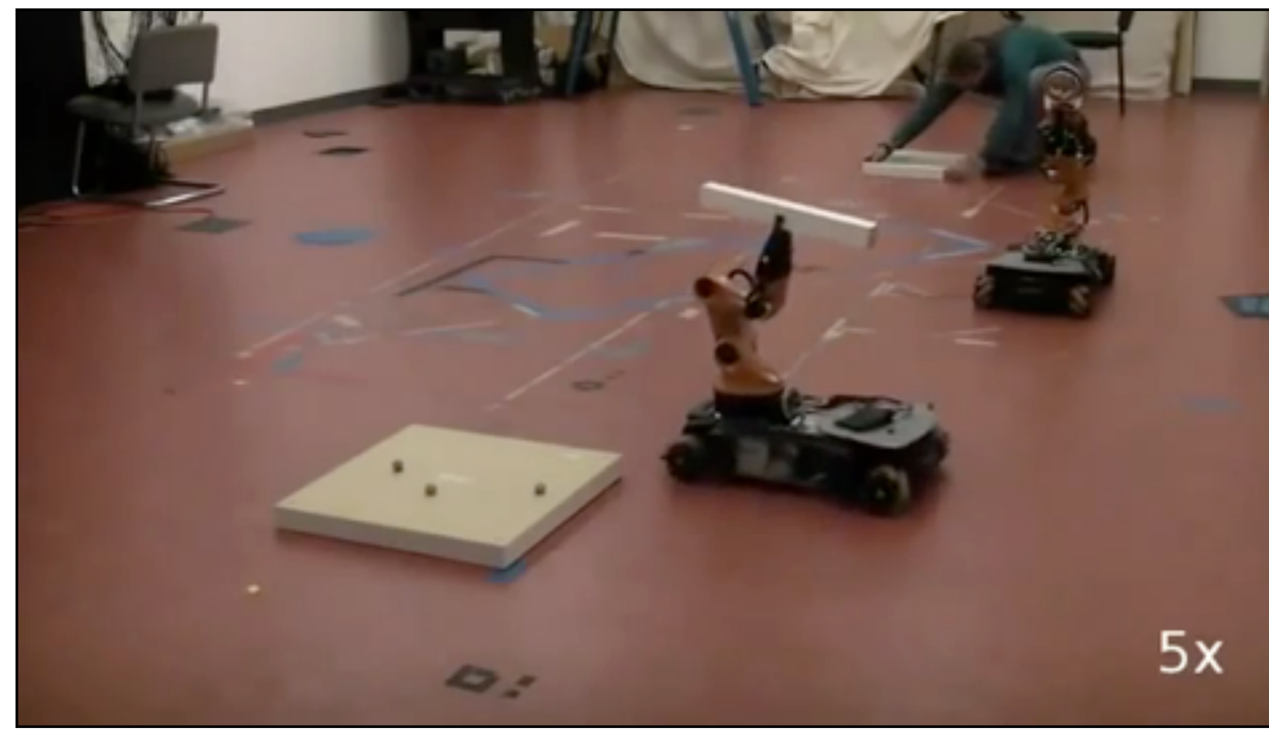
Enable Autonomous Agents to learn to plan effectively in massive stochastic state-spaces.

Minecraft & Robotics

[2]

http://youtu.be/fSLh92zCgIg?t=1m12s

http://youtu.be/B9sYogRVF8Q?t=16s



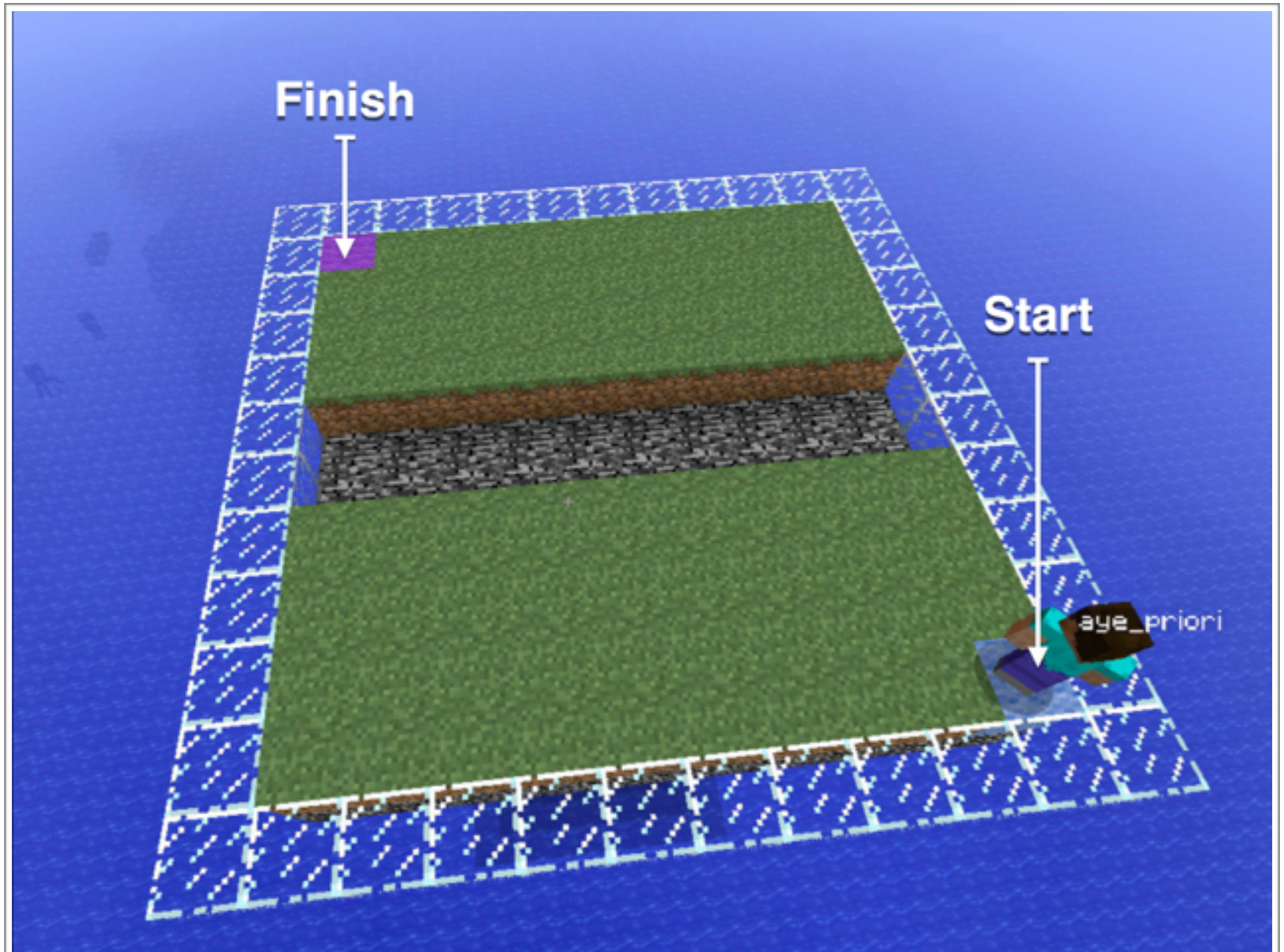
[3] Ross A. Knepper, Todd Layton, John Romanishin, and Daniela Rus

Minecraft

https://vimeo.com/99896931

≈Turing Complete LEGO

Minecraft: The Problem

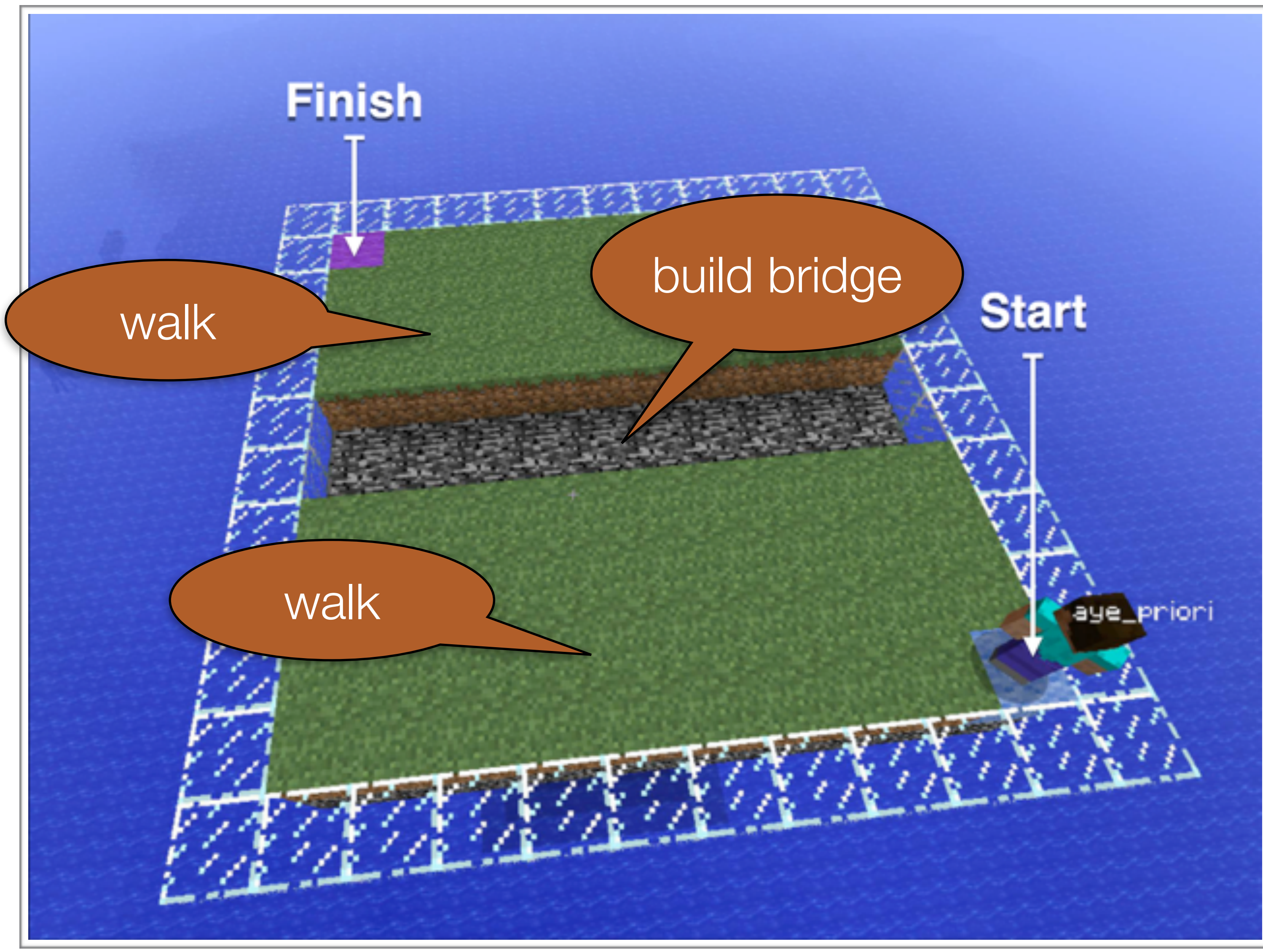


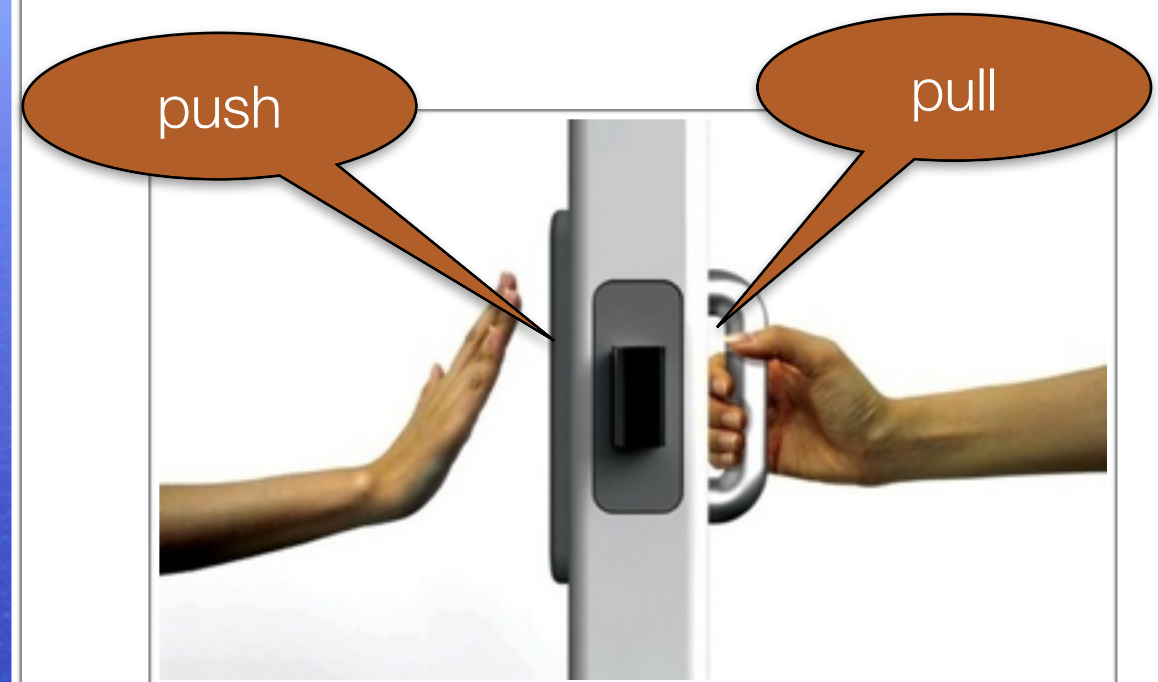
ACTIONS

- Move
- Place
- Destroy
- Use
- Jump
- Rotate
- Look
- Craft
- …

≈ *State Space Size:* 10^{181}

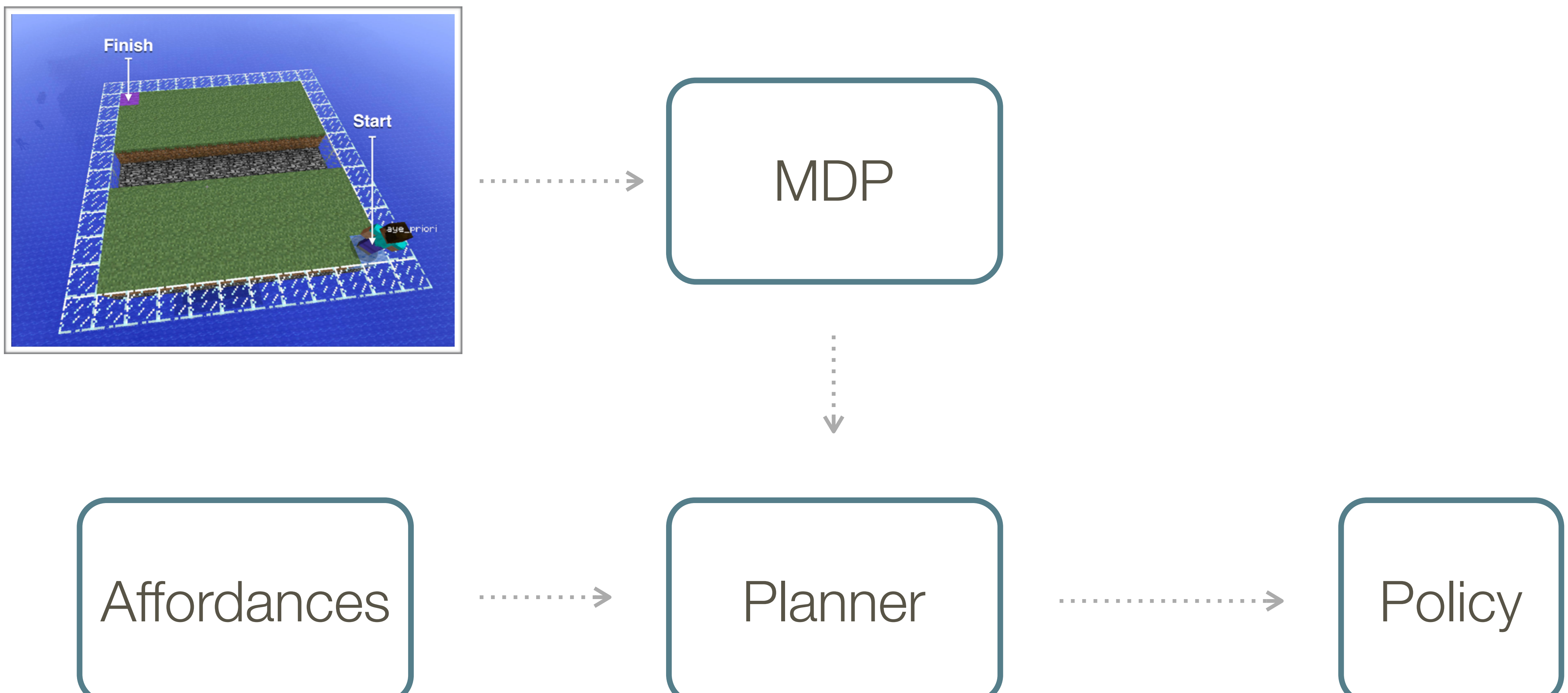
Affordances In Planning





Affordances as knowledge given to an MDP

Idea: Affordances focus the agent on *relevant action possibilities* by pruning irrelevant actions on a state by state basis



Affordances as knowledge added to an MDP

We define an affordance as:

$$\Delta = \langle p, g \rangle \longmapsto \mathcal{A}'$$

Where:

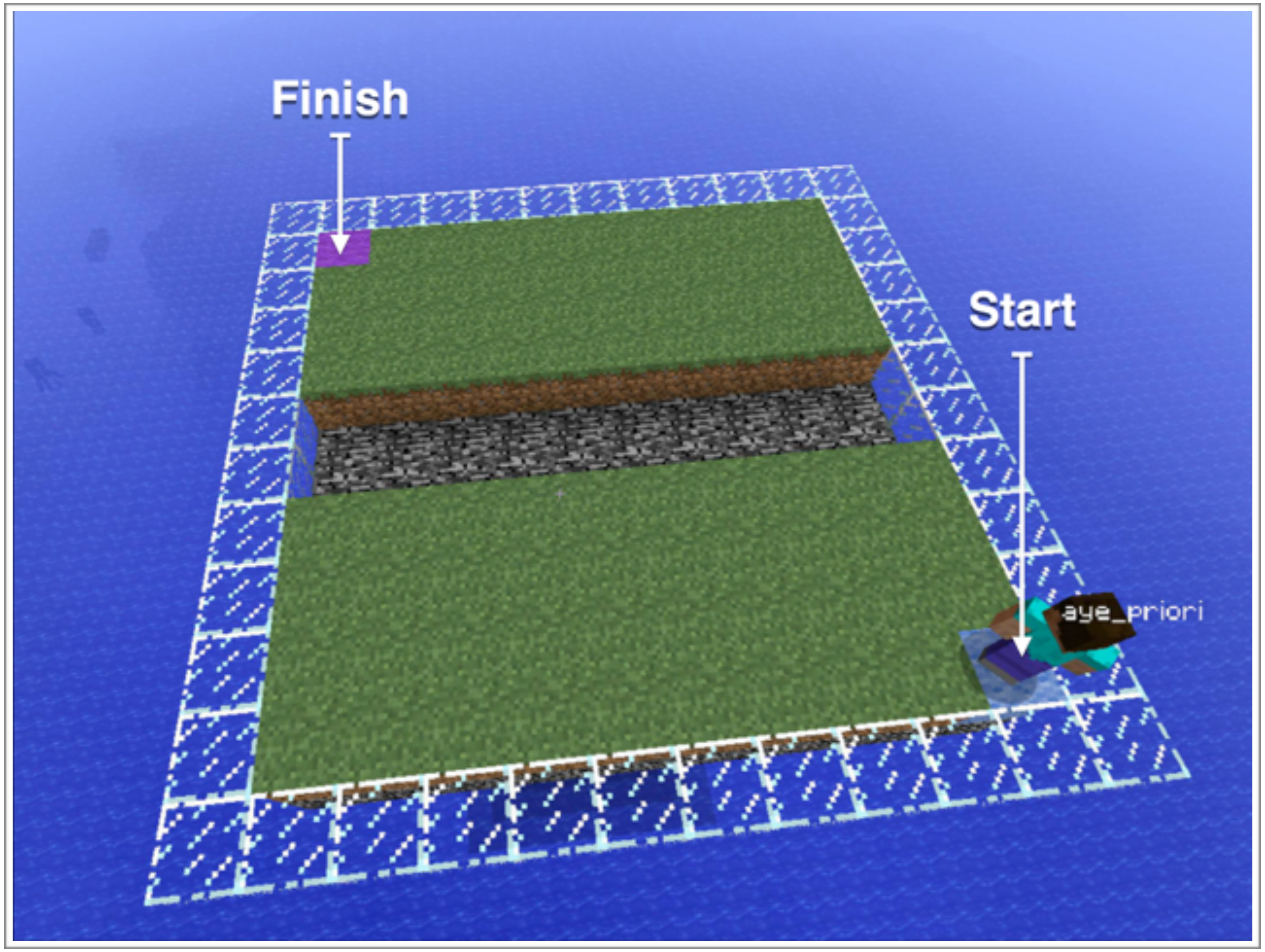
Δ = symbol for an affordance

p = predicate on states

g = lifted goal description

$\mathcal{A}'$ = subset of MDP actions

Affordances Example



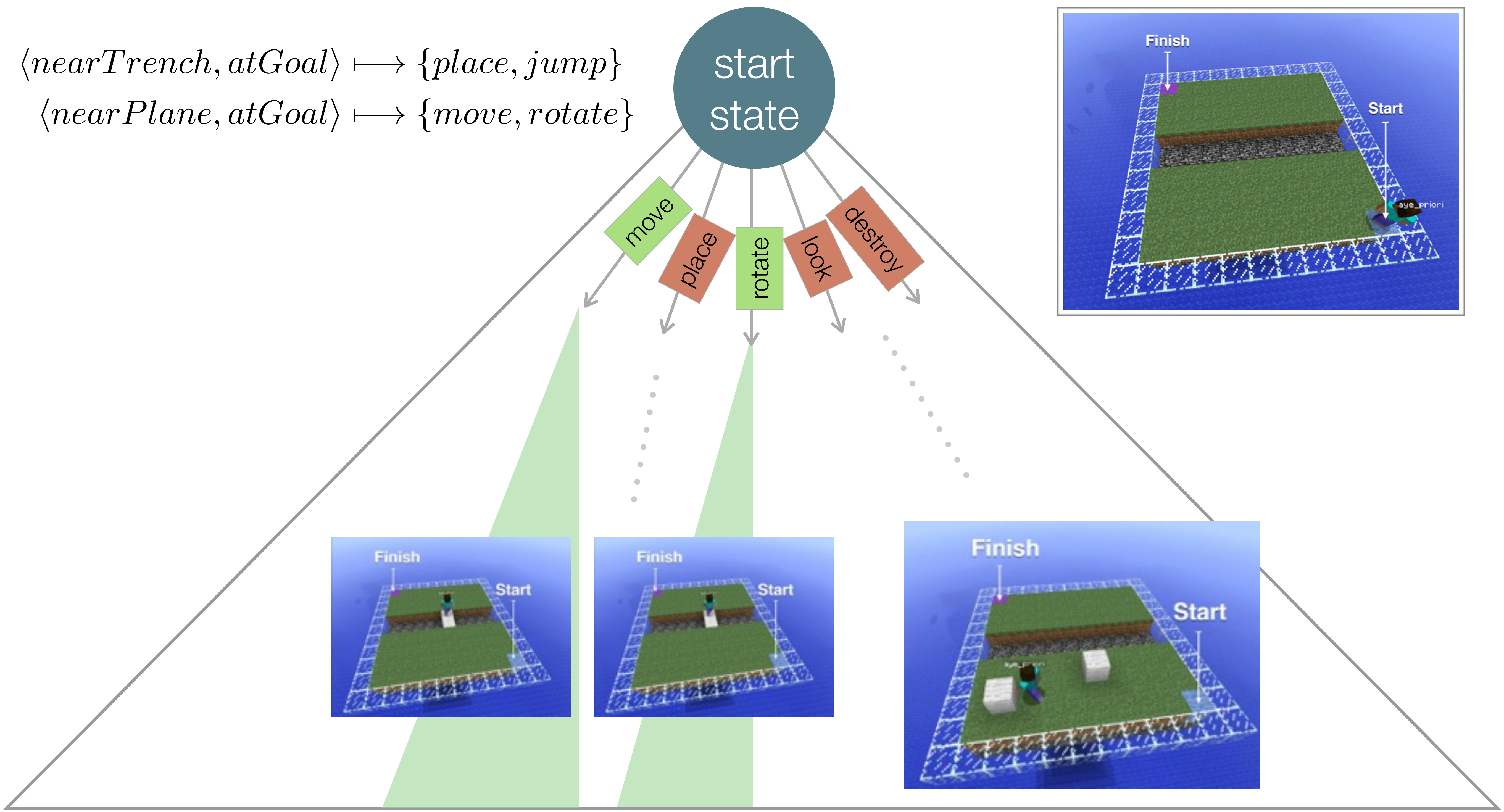
$$\langle nearTrench, atGoal \rangle \longmapsto \{place, jump\}$$

$$\langle nearPlane, atGoal \rangle \longmapsto \{move, rotate\}$$

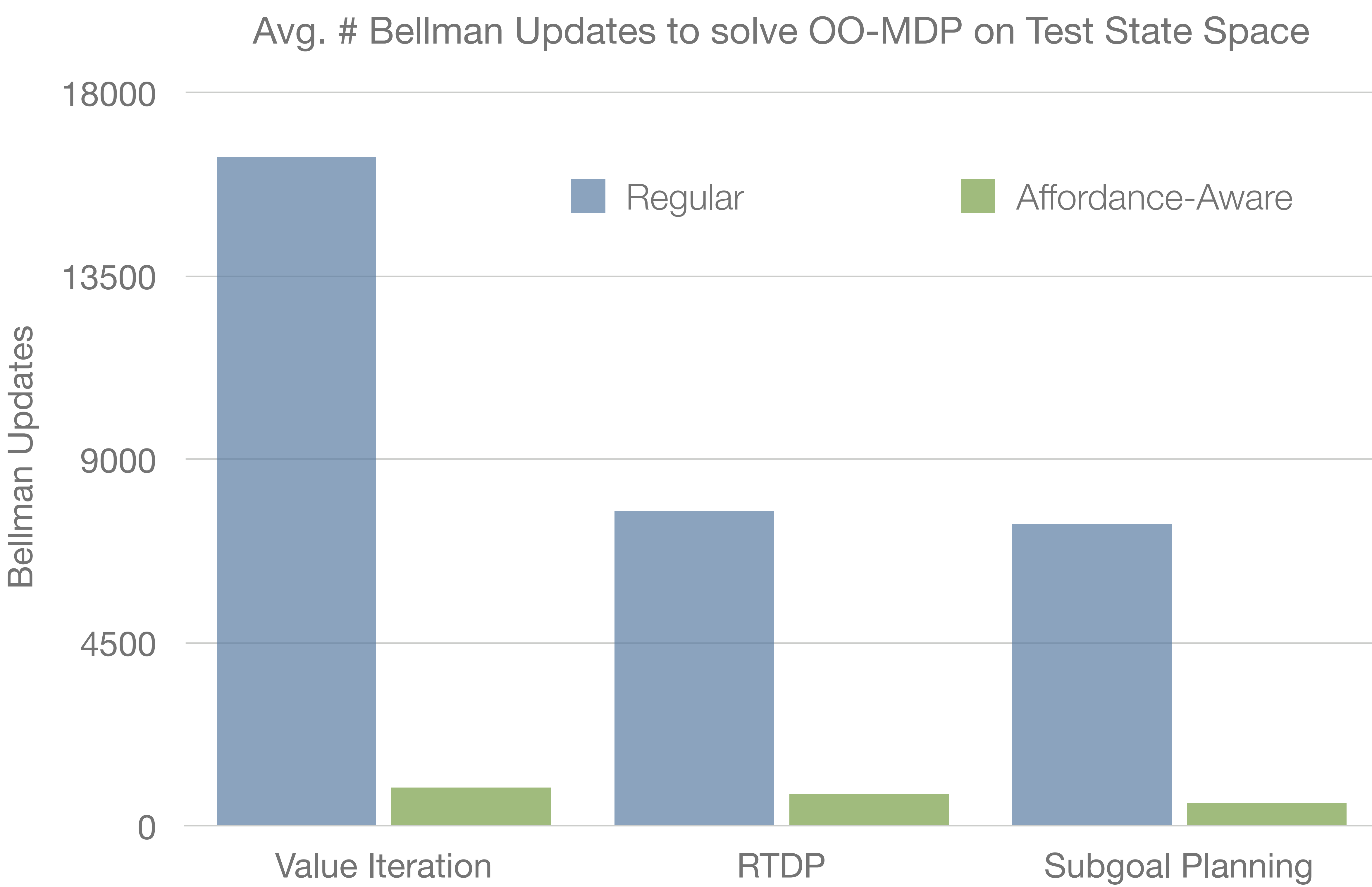
Affordances Example: State-Action Space Pruning

$$\langle nearTrench, atGoal \rangle \longmapsto \{place, jump\}$$

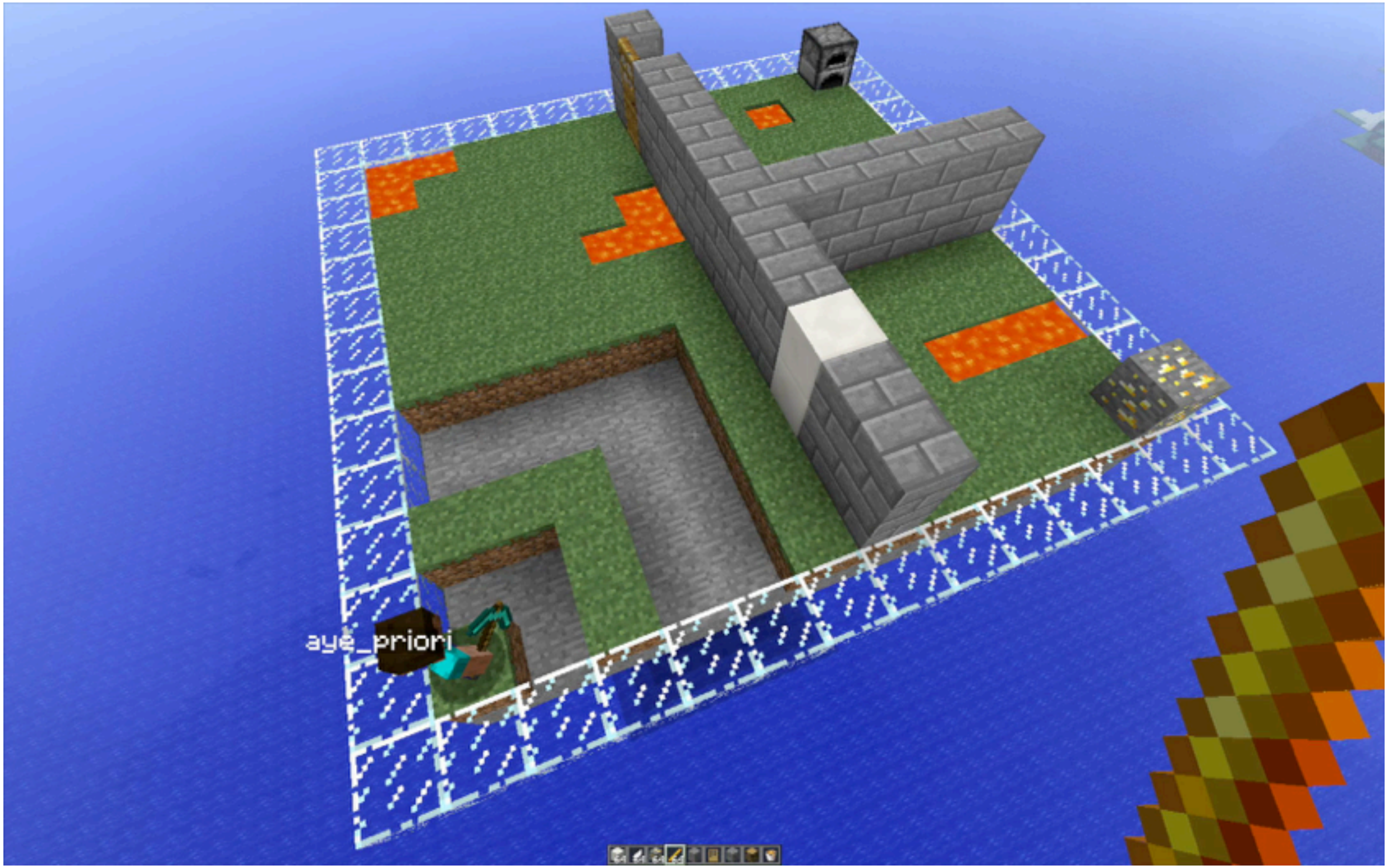
$$\langle nearPlane, atGoal \rangle \longmapsto \{move, rotate\}$$



Sample Results: Expert Affordances



Demo



https://vimeo.com/88689171

Learning Framework: Goal

1. *Compute the probability that an action set contains the optimal action for each reachable state in the MDP*

2. *Learn Dirichlet priors on action sets for each affordance to inform this distribution*

Learning Framework

1. *Compute the probability that an action set contains the optimal action for each reachable state in the MDP*

$$\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) \tag{1}$$

$$= \Pr(\mathcal{A}'_i \cup \ldots \cup \mathcal{A}'_K \mid s, \Delta_1, \ldots, \Delta_K)$$

*assume action sets are disjoint

$$\approx \sum_i^K \Pr(\mathcal{A}'_i \mid s, \Delta_i) \tag{2}$$

Key:

$\mathcal{A}^*$	= candidate set of actions	Δ_i	= the *i*-th affordance
s	= MDP state	$\mathcal{A}'_i$	= the *i*-th affordance's action set

Learning Framework: Goal

2. *Learn Dirichlet priors on action sets for each affordance that inform this distribution*

$$\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) \approx \sum_{i}^{K} \boxed{\Pr(\mathcal{A}_i' \mid s, \Delta_i)}$$

$$\boxed{\Pr(\mathcal{A}_i' \mid s, \Delta_i)} = \Pr(\mathcal{A}_i' \mid n_i, \lambda_i) = \boxed{\Pr(\lambda_i \mid \alpha_i)} \cdot \boxed{\Pr(n_i \mid \beta_i)}$$

Where: $\boxed{\Pr(\lambda_i \mid \alpha_i)} = DirMult(\alpha_i)$ $\boxed{\Pr(n_i \mid \beta_i)} = Dir(\beta_i)$

Key:

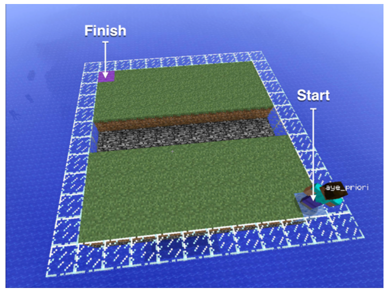
$\mathcal{A}^*$ = candidate set of actions	Δ_i = the *i*-th affordance
s = MDP state	$\mathcal{A}_i'$ = the *i*-th affordance's action set
λ_i = multinomial over actions	n_i = multinomial over action set size
α_i = Dirichlet parameters	β_i = Dirichlet Multinomial parameters

Full Learning Process

```
Input:    P (set of predicates), G (set of goal descriptions)
```

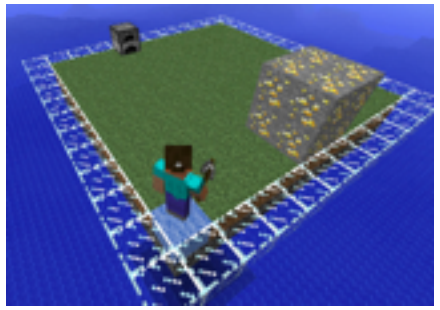
RANDOMLY GENERATE SMALL WORLDS:

AtGoal



. . . . . .

MakeGold



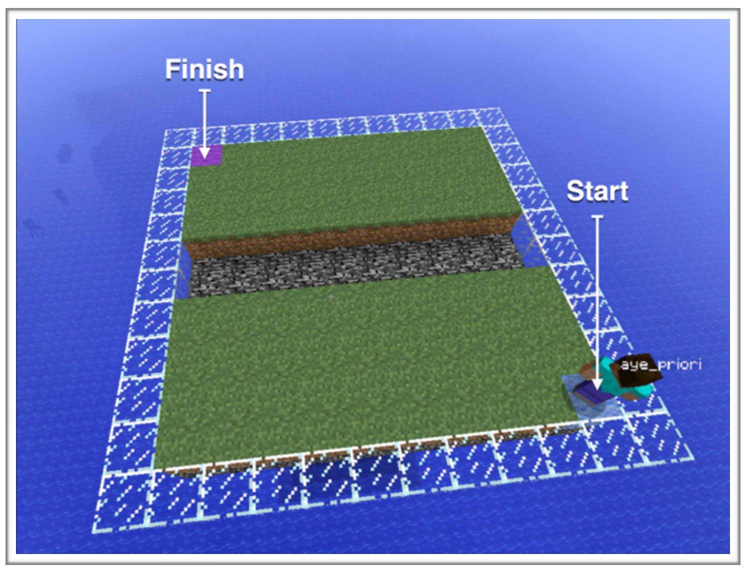
SOLVE EACH MDP

π_1 . . . . . . π_N

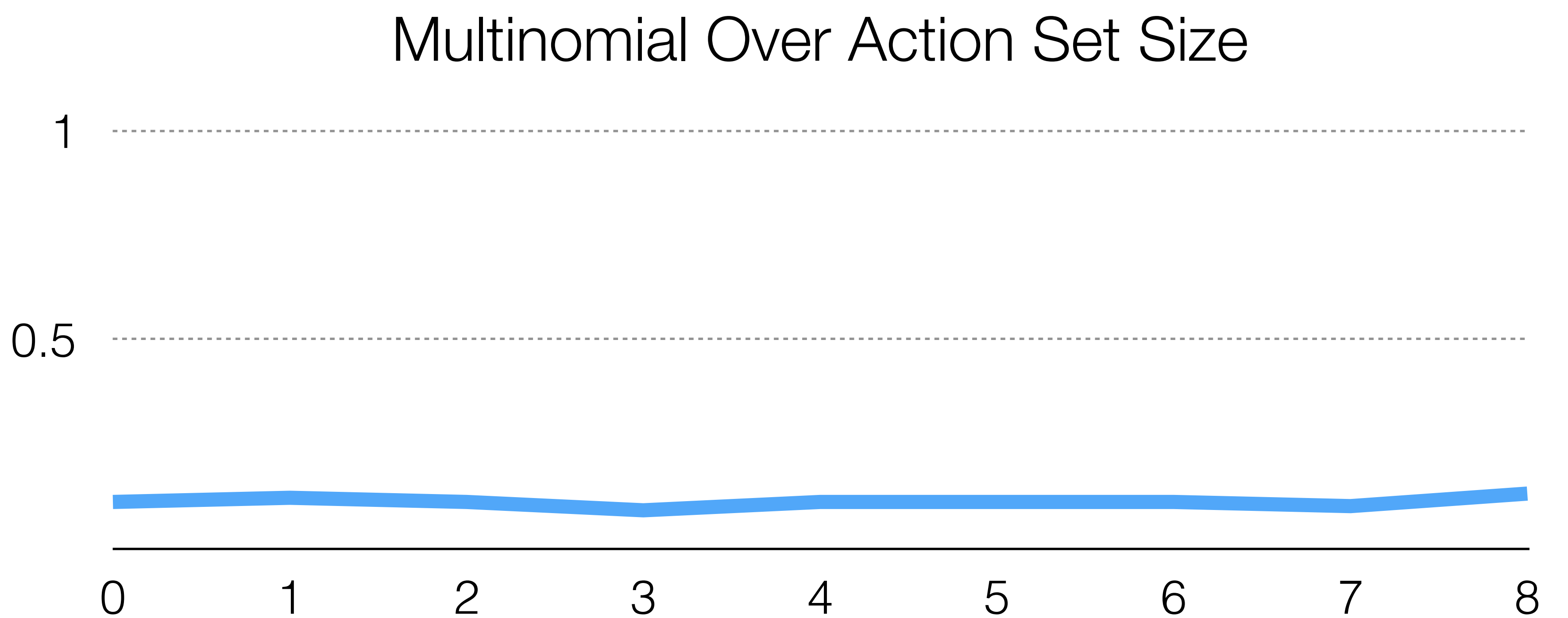
COUNT:

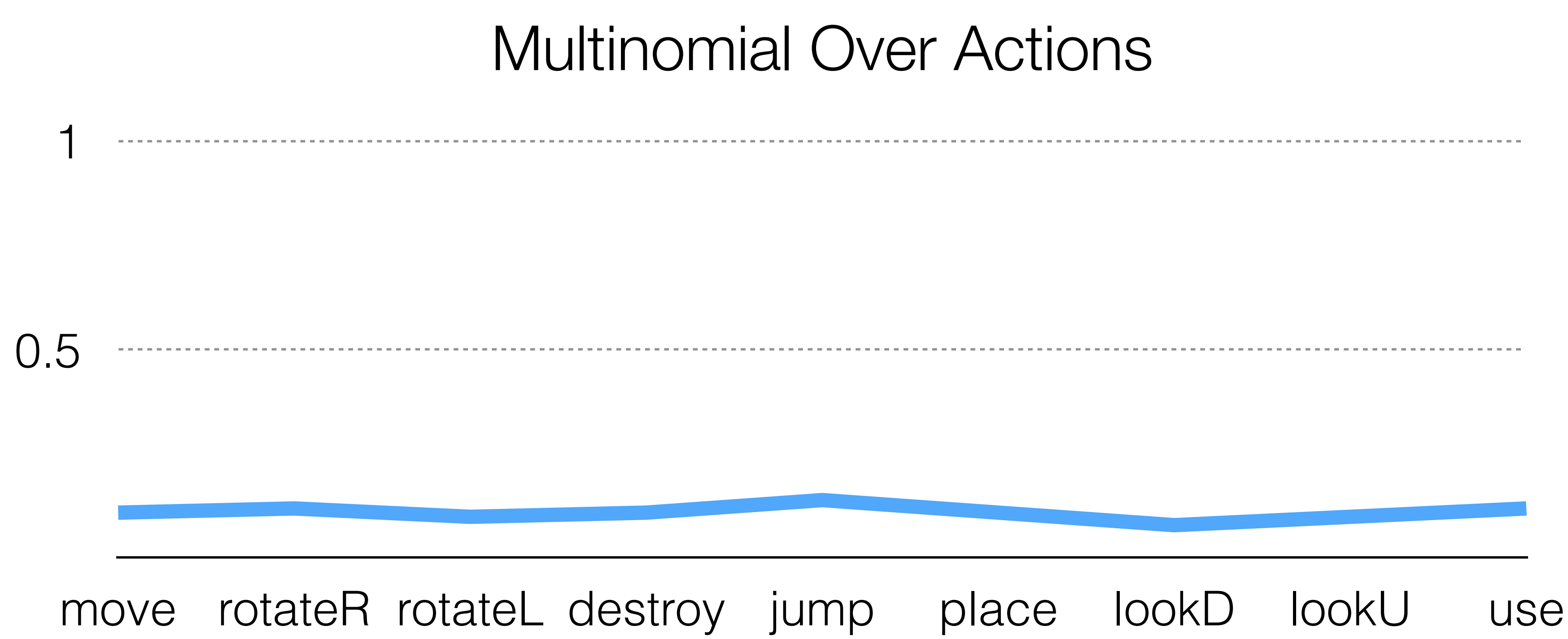
$$\Delta_i.parameterUpdate(\pi_1, \ldots, \pi_N)$$

Learning Example: Before Learning

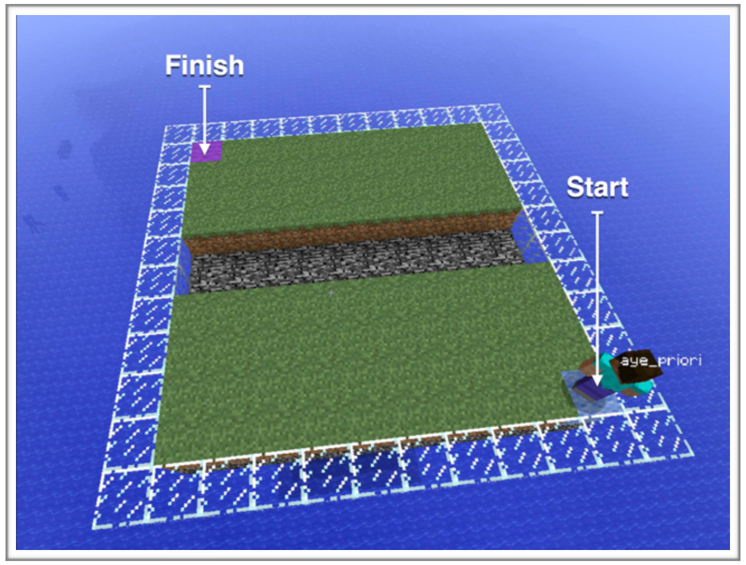


p = nearTrench
g = AtGoal

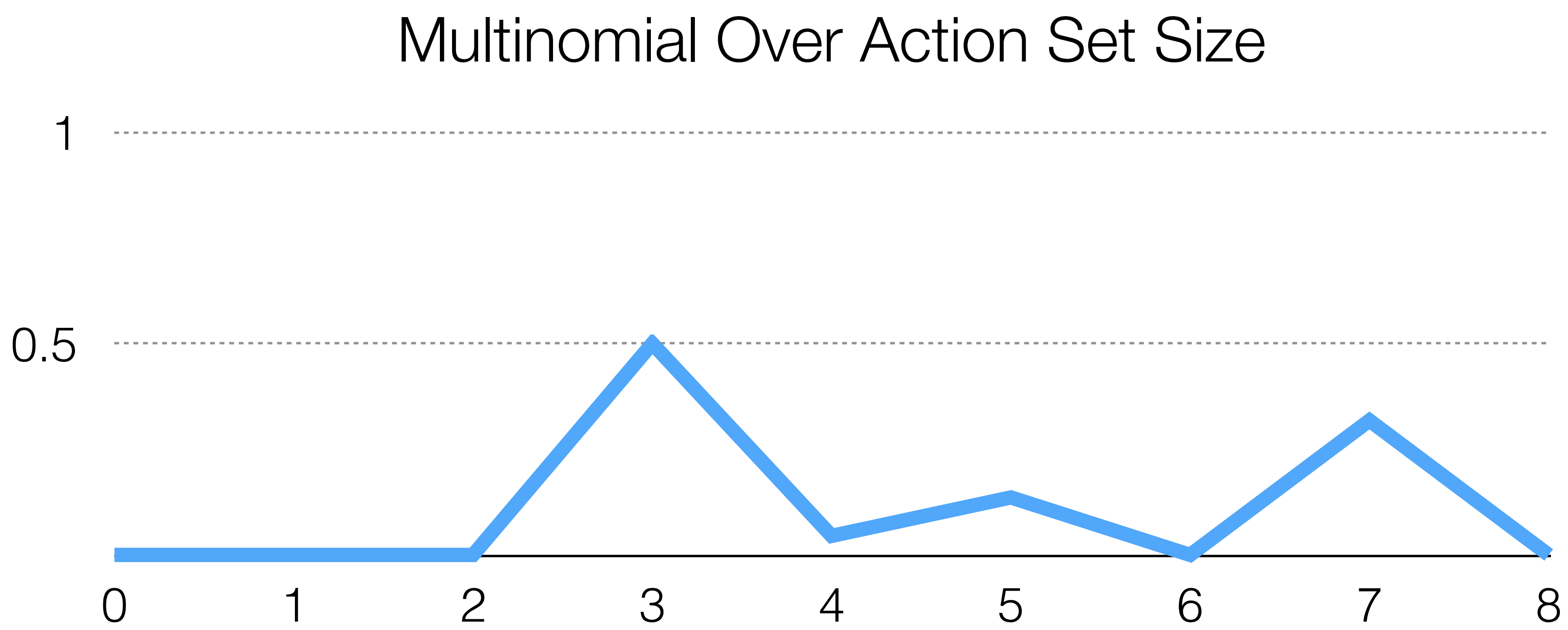


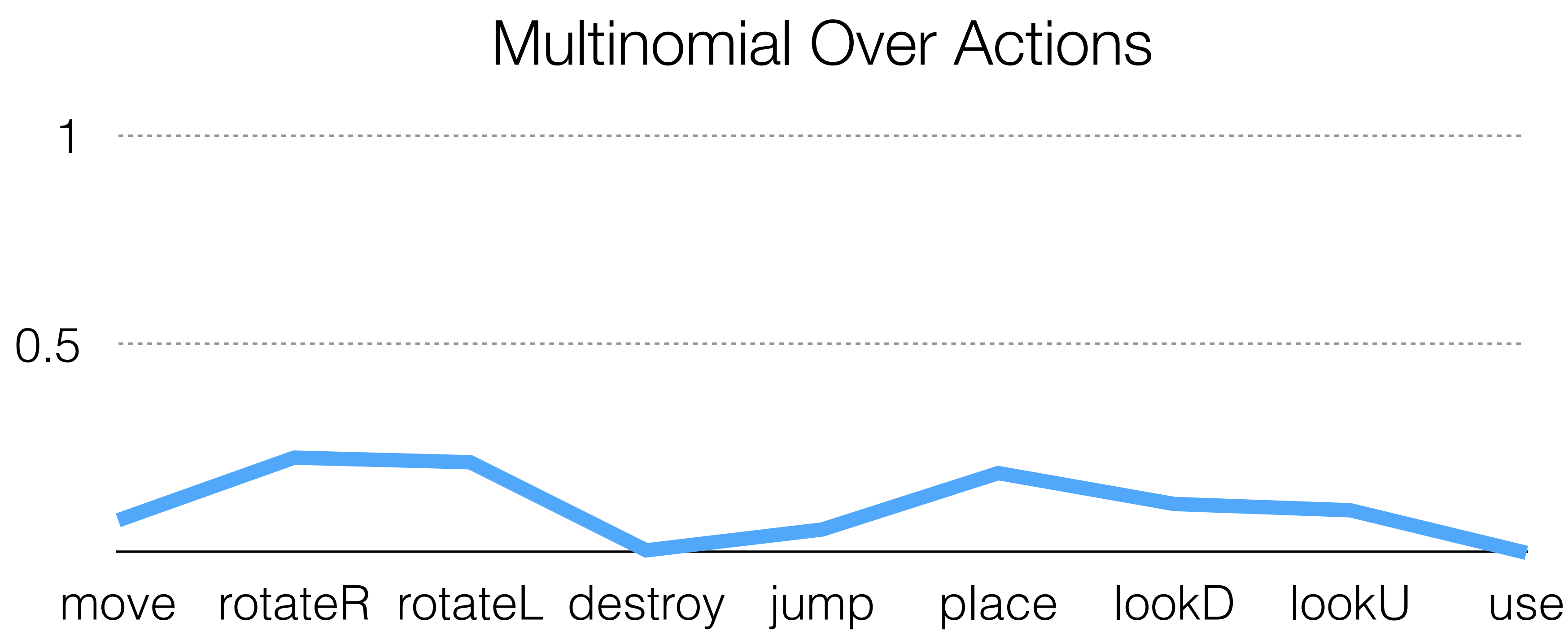


Learning Example: After Learning



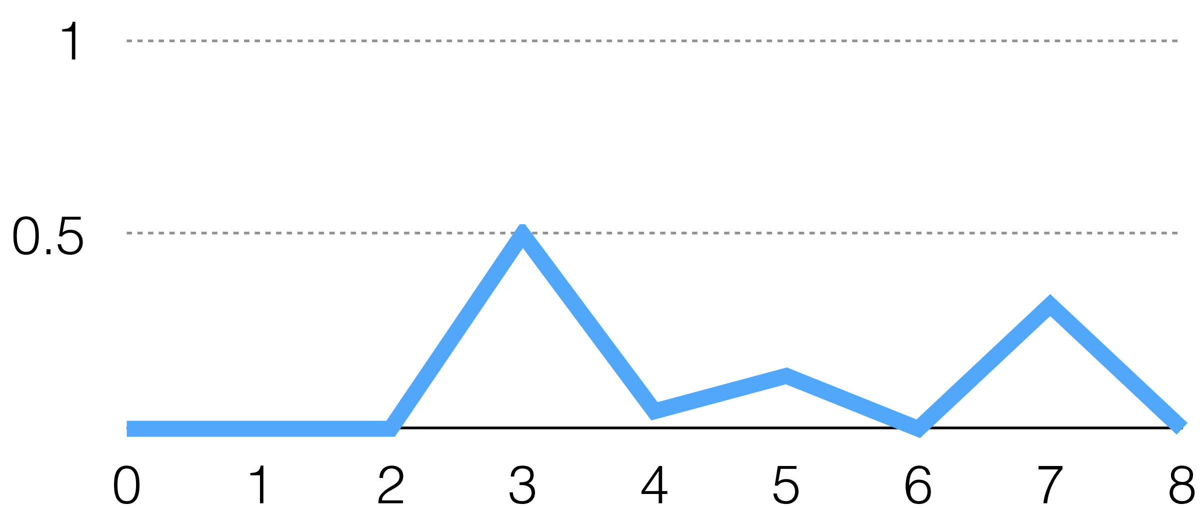
p = nearTrench
g = AtGoal





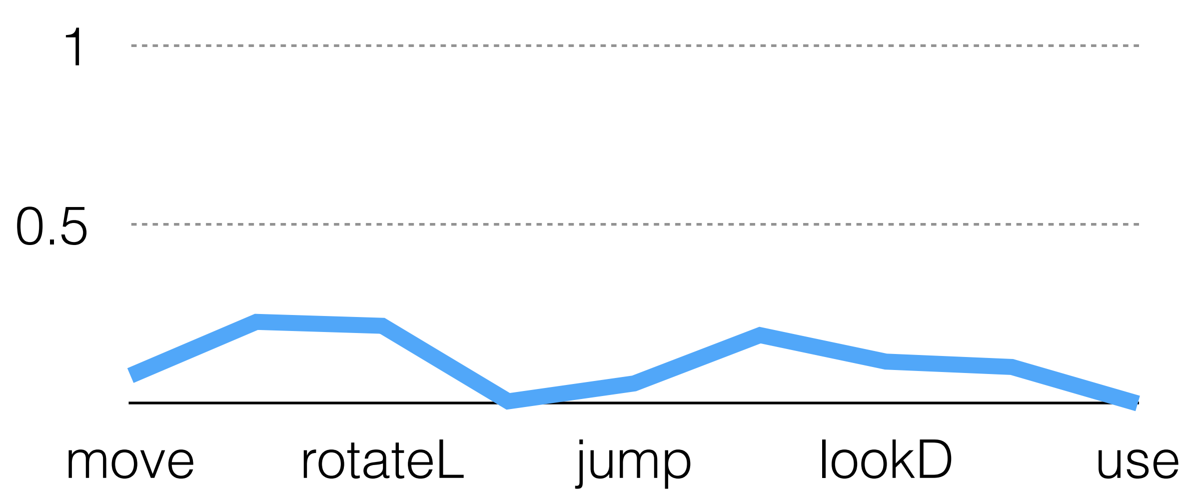
Learning Example

Multinomial Over Action Set Size



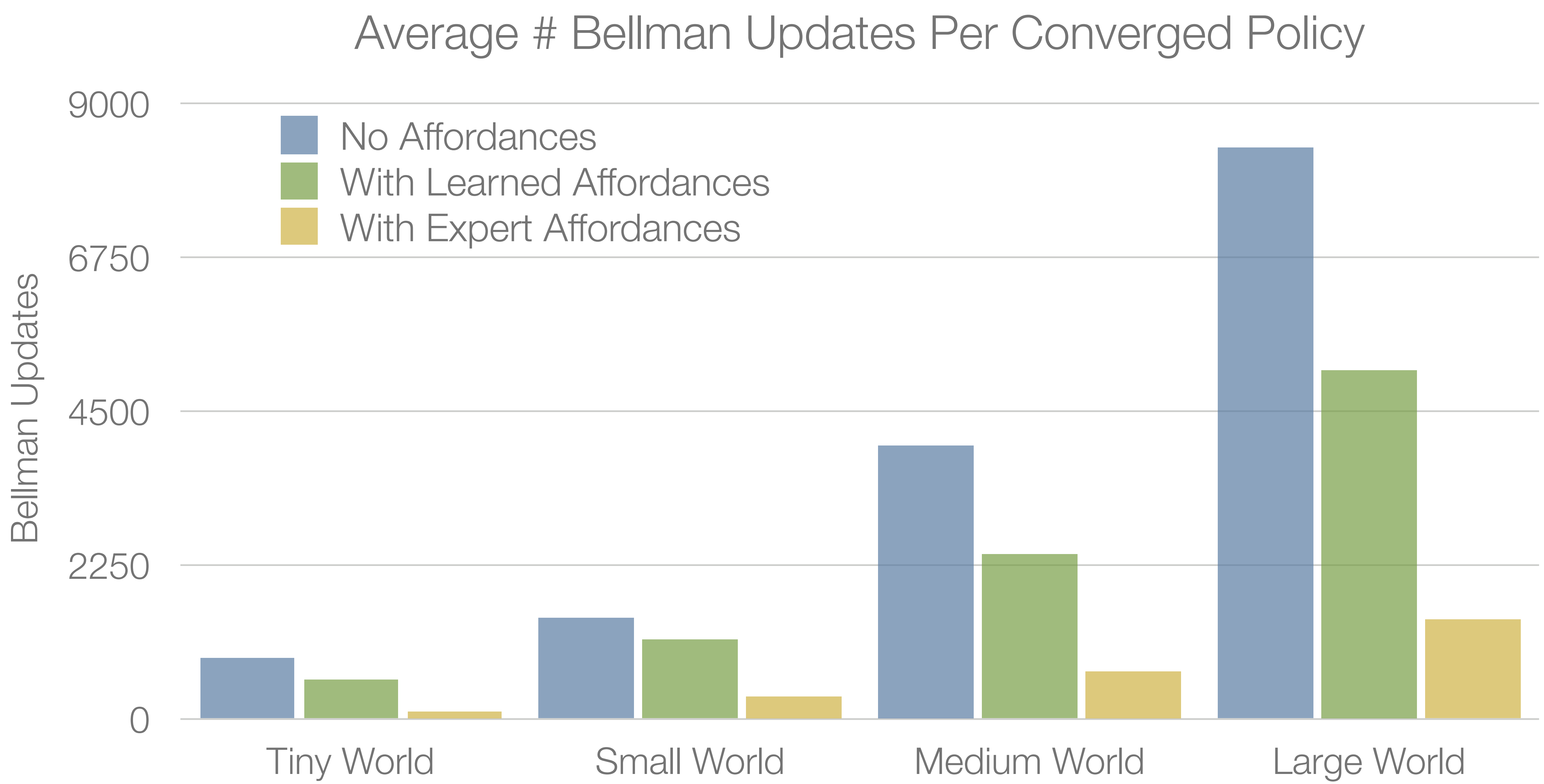
Sample → ***n***

Multinomial Over Actions



*Take **n** unique samples* → ***A***

Learning Results



Related Work

- Temporarily Extended Actions [5, 6, 7]
- Temporal Logic [8, 9]
- Action Pruning [11, 12]
- Heuristics [13, 14]

Future Work

- Extending predicates to logical expressions
- Incorporate more of the Minecraft Domain
- Learning High-level Representations
- Deploy on robots, other domains (cooking, javascript, Atari)

Contributions

- Defined affordance as formal knowledge added to an MDP.
- Realized speedups for planning in large stochastic state spaces.
- Demonstrated framework for learning affordances from interaction in the world.

References

[1] Jennifer L. Barry, DAART ROS package, http://wiki.ros.org/darrt

[2] https://www.youtube.com/watch?v=B9sYogRVF8Q

[3] Ross A. Knepper, Todd Layton, John Romanishin, and Daniela Rus. “IkeaBot: An Autonomous Multi-Robot Coordinated Furniture Assembly System”. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, Germany, May 2013.

[4] http://qph.is.quoracdn.net/main-qimg-f0a341a110341f5a58a93b75b491448d

[5] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of markov decision processes using macro-actions. In *Proceed- ings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 220–229. Morgan Kaufmann Publishers Inc., 1998.

[6] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.

[7] Nicholas K. Jong. The utility of temporal abstraction in reinforcement learning. In Proceedings of the Seventh Interna- tional Joint Conference on Autonomous Agents and Multiagent Systems, 2008.

[8] Fahiem Bacchus and Froduald Kabanza. Using temporal logic to control search in a forward chaining planner. In *In Proceedings of the 3rd European Workshop on Planning*, pages 141–153. Press, 1995.

[9] Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116:2000, 1999.

[10] A.A. Sherstov and P. Stone. Improving action selection in mdp’s via knowledge transfer. In *Proceedings of the 20th national conference on Artificial Intelligence*, pages 1024–1029. AAAI Press, 2005.

[11] Benjamin Rosman and Subramanian Ramamoorthy. What good are actions? accelerating learning using learned action priors. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[12] Eric A Hansen and Shlomo Zilberstein. Solving markov decision problems using heuristic search. In Proceedings of AAAI Spring Symposium on Search Techniques from Problem Solving under Uncertainty and Incomplete Information, 1999.

[13] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy in- variance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999

Extra Slides: Learning Math

Probability that an action set contains the optimal action for each reachable state in the MDP

$$\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K)$$

approximate assuming action sets are disjoint

$$\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) \approx \sum_i^K \Pr(\mathcal{A}_i' \mid s, \Delta_i)$$

For each affordance

$$\Pr(\mathcal{A}_i' \mid s, \Delta_i) = \Pr(\mathcal{A}_i' \mid n_i, \lambda_i) = \Pr(\lambda_i \mid \alpha_i) \cdot \Pr(n_i \mid \beta_i)$$

$$\Pr(\lambda_i \mid \alpha_i) = DirMult(\alpha_i)$$

$$\Pr(n_i \mid \beta_i) = Dir(\beta_i)$$

Key:

$\mathcal{A}_i' = \text{Affordance}_i$ action set
$\mathcal{A}^* =$ Candidate action set
$\lambda =$ Actions sampled from affordance-specific multinomial
$n =$ Action set size sampled from affordance-specific multinomial
$\Delta_i = \text{Affordance}_i$
$\alpha =$ Dirichlet Multinomial parameters for prior on actions
$\beta =$ Dirichlet parameters for prior on action set size
$s =$ MDP state

Extra Slides: Graphical Model

Key:

$\mathcal{A}'_i =$ Affordance$_i$ action set
$\mathcal{A}^* =$ Candidate action set
$\lambda =$ Actions sampled from affordance-specific multinomial
$n =$ Action set size sampled from affordance-specific multinomial
$\Delta_i =$ Affordance$_i$
$\alpha =$ Dirichlet Multinomial parameters for prior on actions
$\beta =$ Dirichlet parameters for prior on action set size
$s =$ MDP state