Competence-Preserving Retention of Learned Knowledge in Soar's Working and Procedural Memories

Nate Derbinsky

John E. Laird

University of Michigan

support by

Motivation

Goal. Long-term modeling of human-level behavior.

Problem. <u>Extended tasks</u> that involve amassing <u>large amounts of knowledge</u> can <u>lead to performance degradation</u> in existing systems (e.g. Kennedy & Trafton '07; Douglass et al. '09).

Common Approach

Forgetting. Selective retention of learned knowledge.

Challenge. Balance…

– maintenance of high model competence &

– reduction of computational resources

across a variety of tasks.

This Work

Hypothesis. Useful to forget a memory if…

1. not useful (via *base-level activation*) &
2. likely can *reconstruct* if necessary

Evaluation. 2 complex tasks, 2 memories (Soar)

Mobile Robot Navigation

Working Memory
- bounds decision time
- completes task
  - 1 hour

Multi-Player Dice

Procedural Memory
- 50% memory reduction
- competitive play
  - days

Task Independent

Related Work

Forgetting

Modeling humans

- (e.g. Anderson et al. ‘96; Chong ‘03, ‘04)

Cognitive benefits of forgetting

- (e.g. Altmann & Gray ’02; Schooler & Hertwig ‘05)

Computational scaling

- (Kennedy & Trafton ‘07): internal tasks, recency

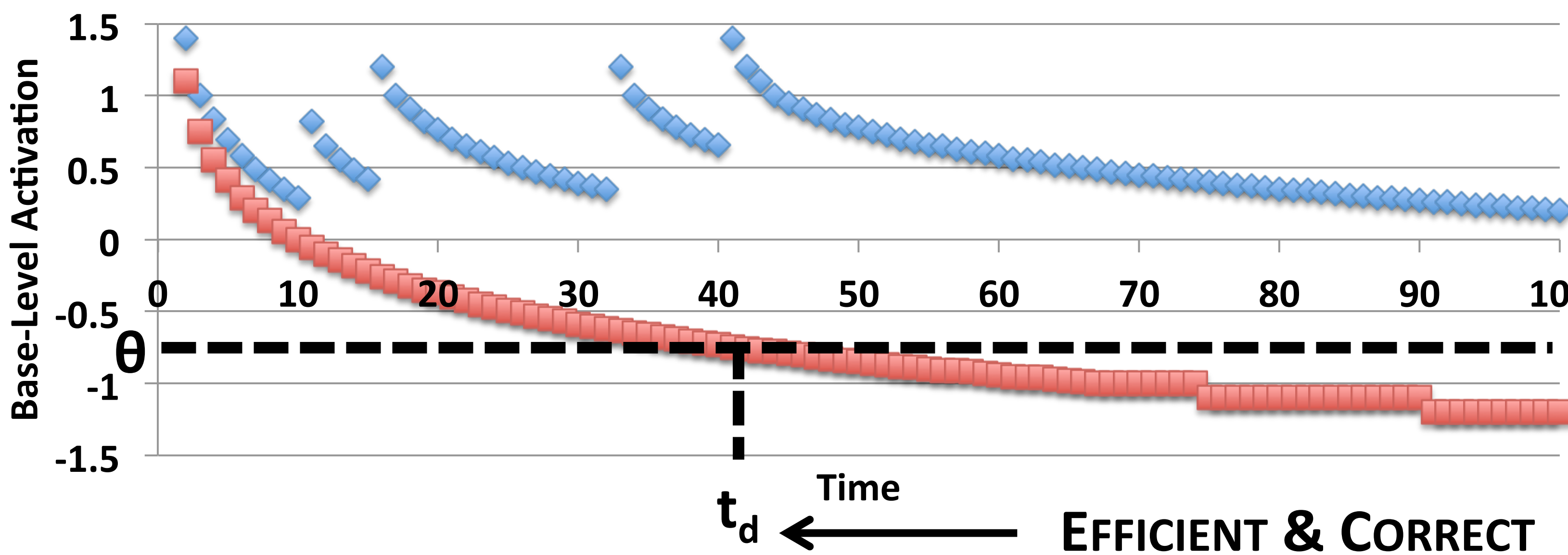
Related Work

Forgetting via Base-Level Activation (ICCM ’12b)

Base-Level Activation (Anderson et al. ‘04)

$$\ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$

- Predict future memory usage via history
- Core to ACT-R declarative module
  - Models retrieval bias, errors, and forgetting via failure



EFFICIENT & CORRECT

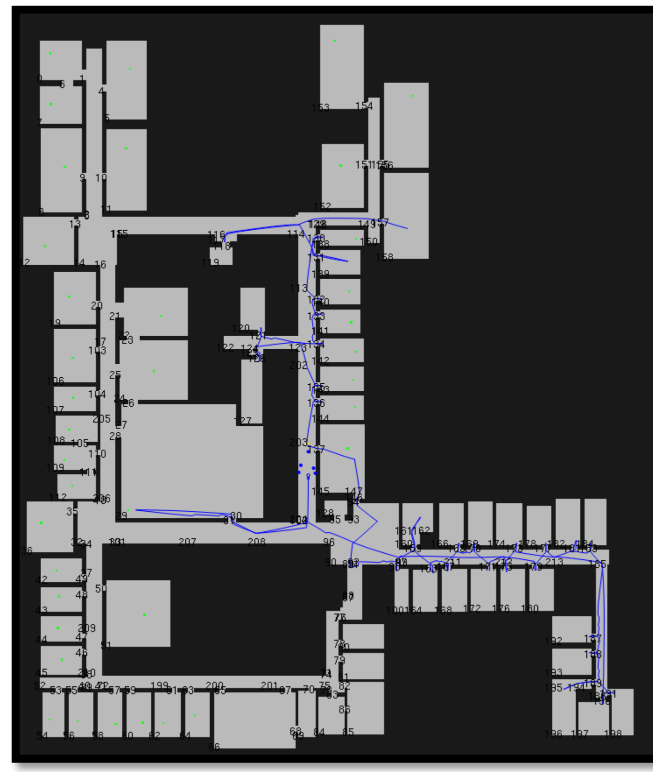
~~*O*(# memories)~~

Task #1: Mobile Robotics
(Laird, Derbinsky & Voigt ’11)

Simulated Exploration & Patrol

– 3rd floor, BBB Building, UM
  - 110 rooms
  - 100 doorways

– Builds map in memory from experience



Problem: Decision Time

Issue. Large working memory

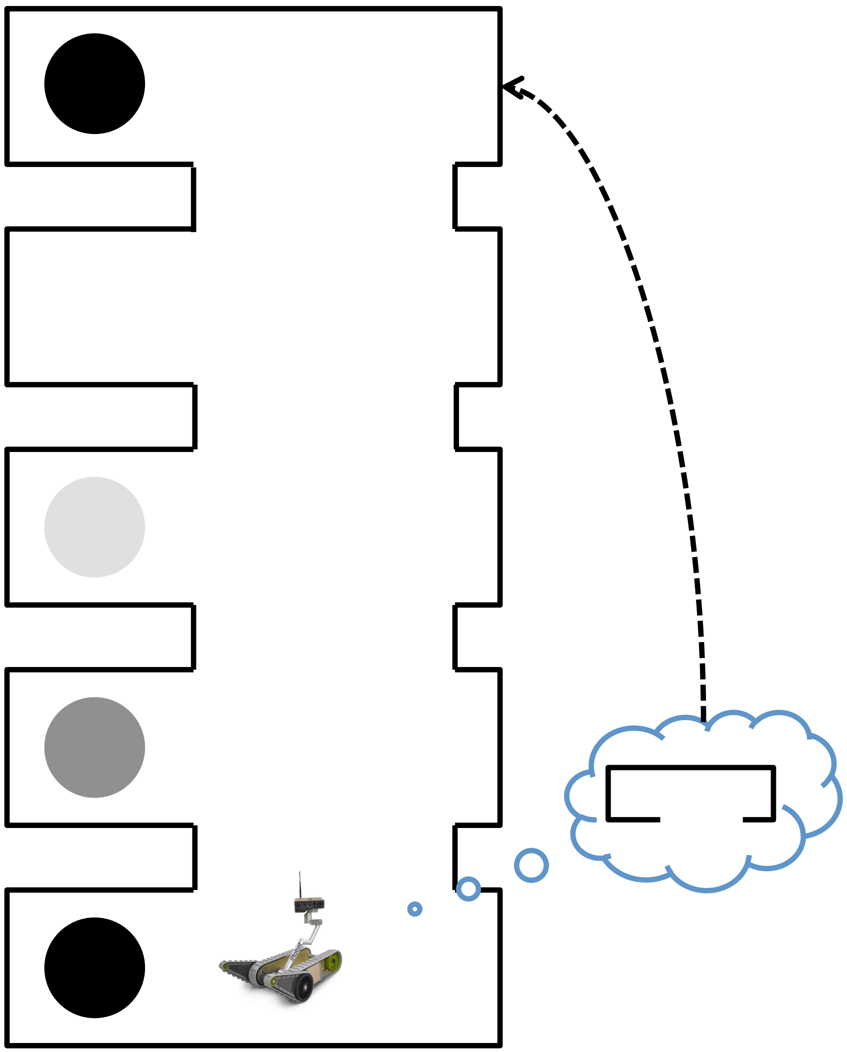
- Minor: rule matching (Forgy 1982)
- Major: episodic reconstruction (Derbinsky & Laird ’09)
  - |episode|~|working memory|

Forgetting Policy. Memory hierarchy

1. Forget unused short-term features of long-term objects
2. Retrieve from LTM as necessary

Task Independent

Map Knowledge



Room Features

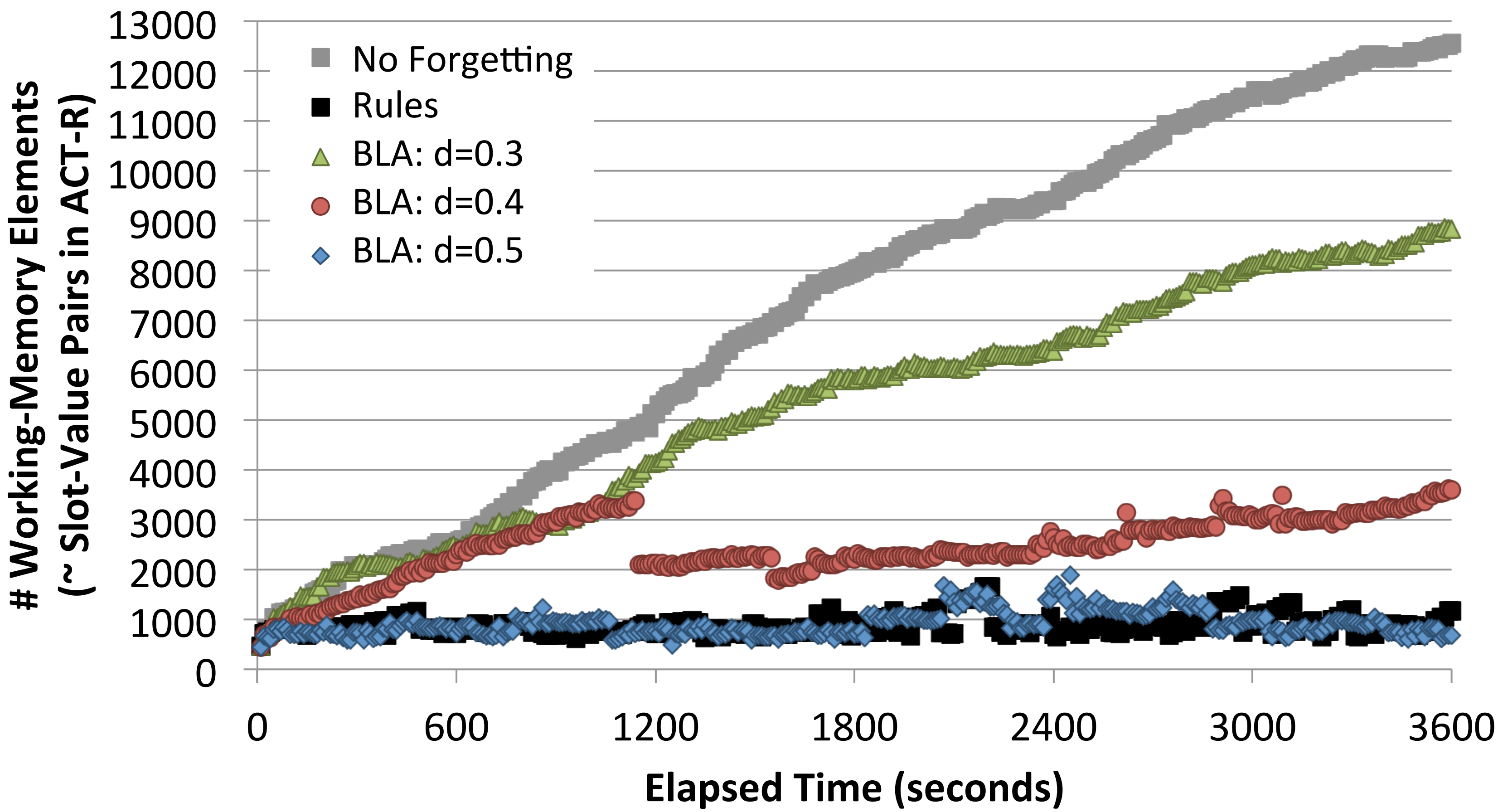
- Position, size
- Walls, doorways
- Objects
- Waypoints

Usage

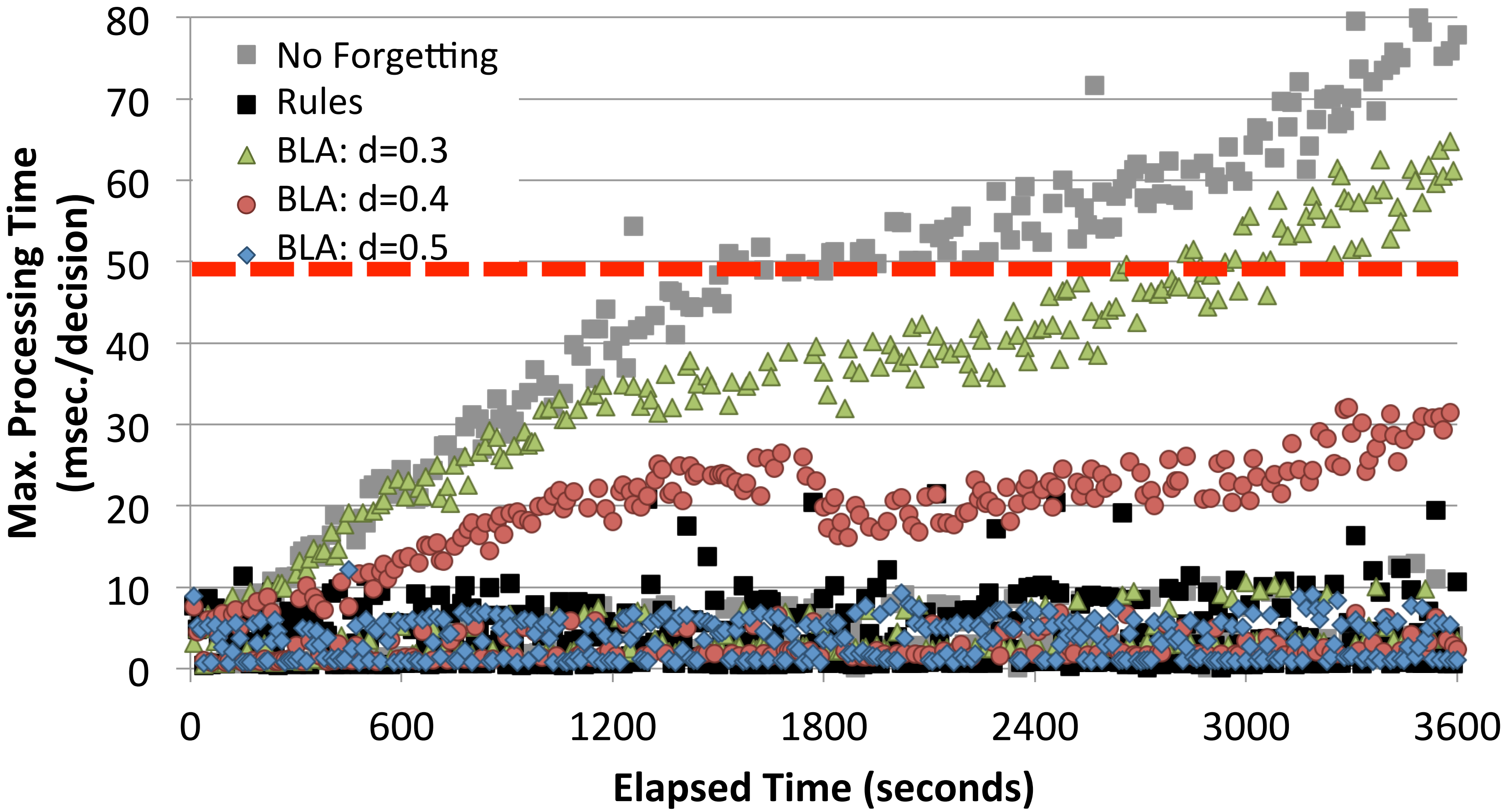
- Exploration (-->LTM)
- Planning/navigation (<--LTM)

  Reconstruction

Results: Working-Memory Size



Results: Decision Time

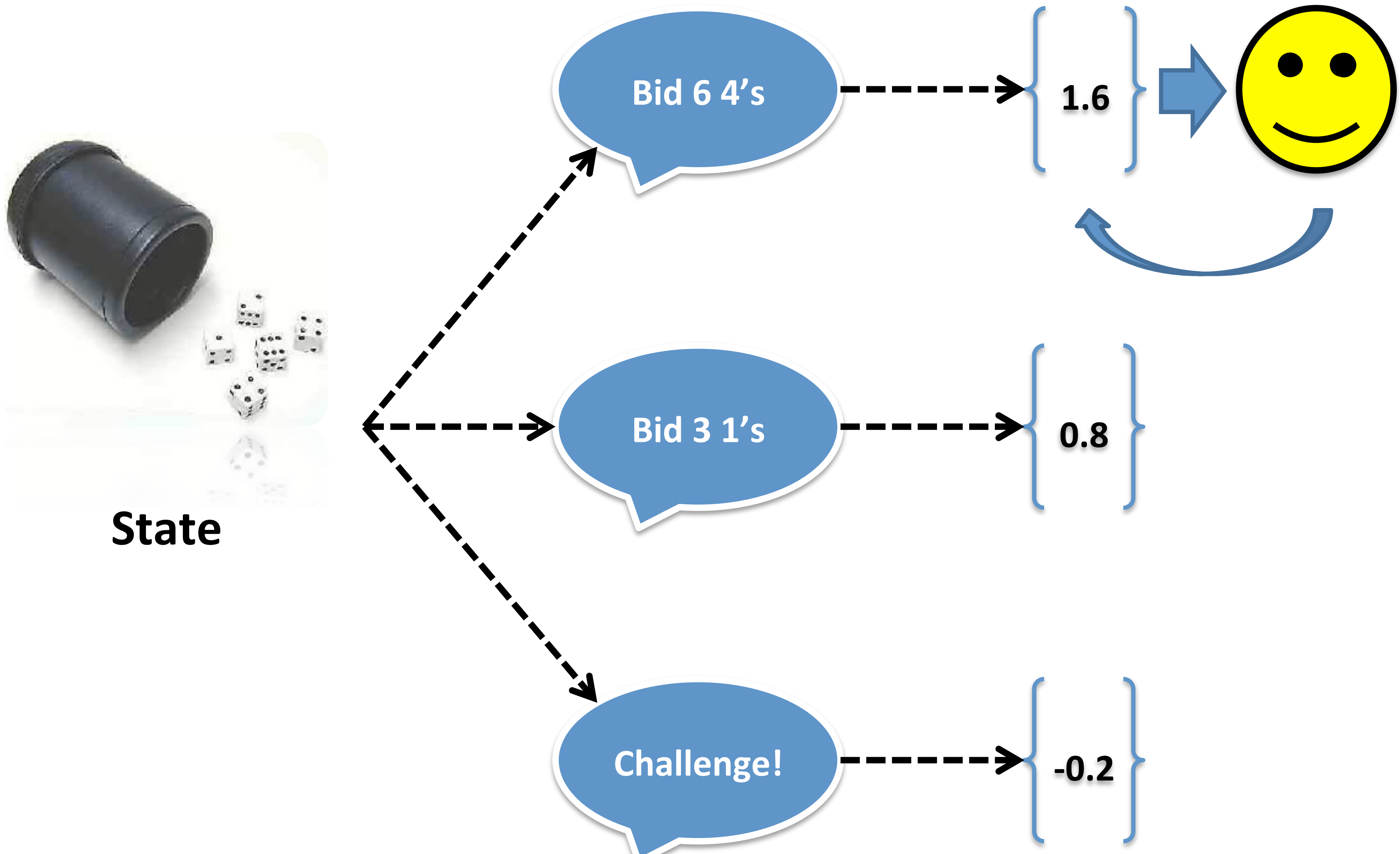


Task #2: Liar's Dice

(Laird, Derbinsky & Tinkerhess '11)

- Complex rules, hidden state, stochasticity
  - Rampant uncertainty
- Model learns via reinforcement learning (RL)
  - Large state space (10^6-10^9 for 2-4 players)

Reasoning --> Action Knowledge



Problem: Memory Consumption

Issue. RL value-function representation: (s,a)->#

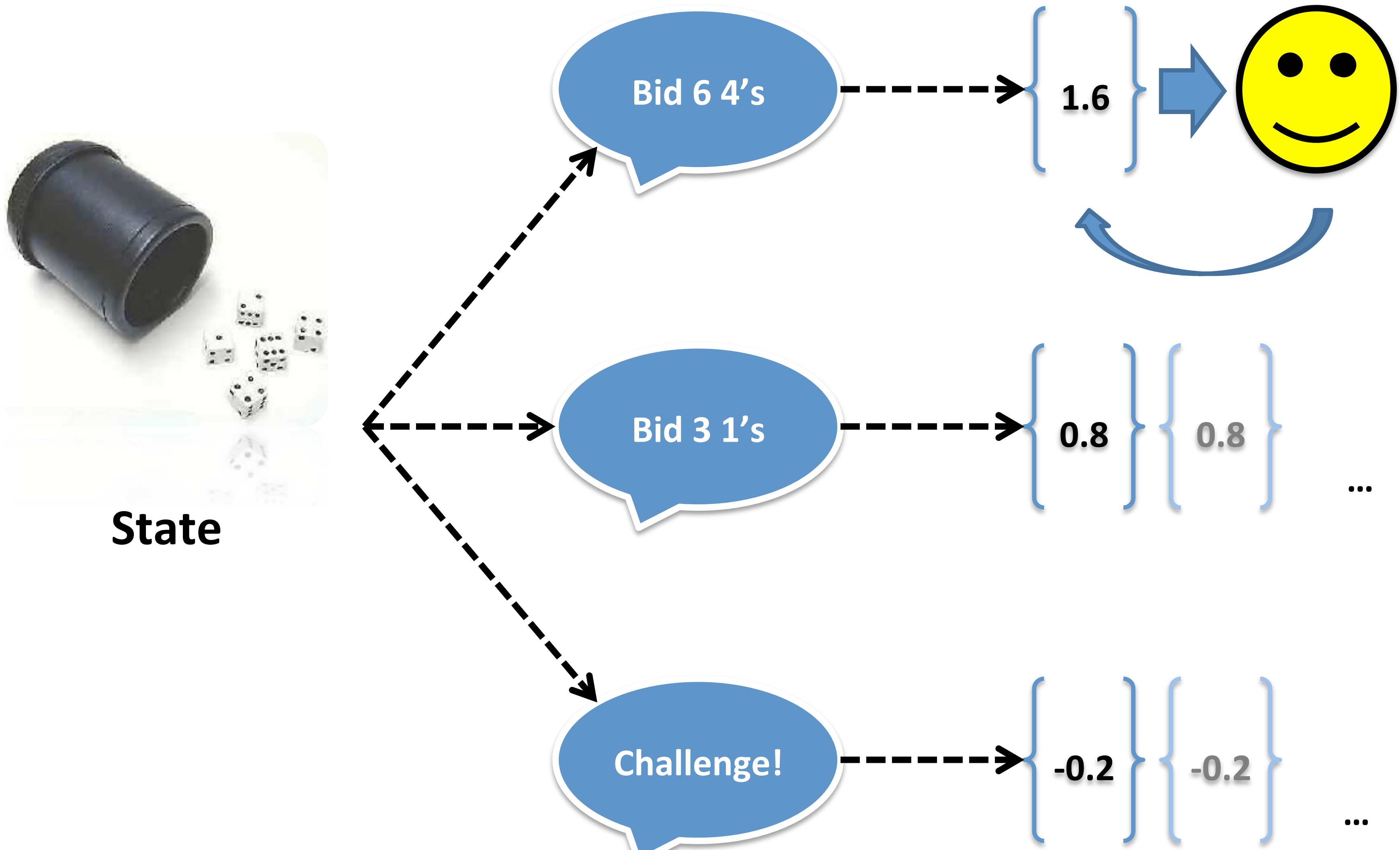
- Soar: procedural knowledge (*RL* rules)
- Many possible actions per turn;
  at most feedback for a single action

Forgetting Policy. Keep what you can't reconstruct

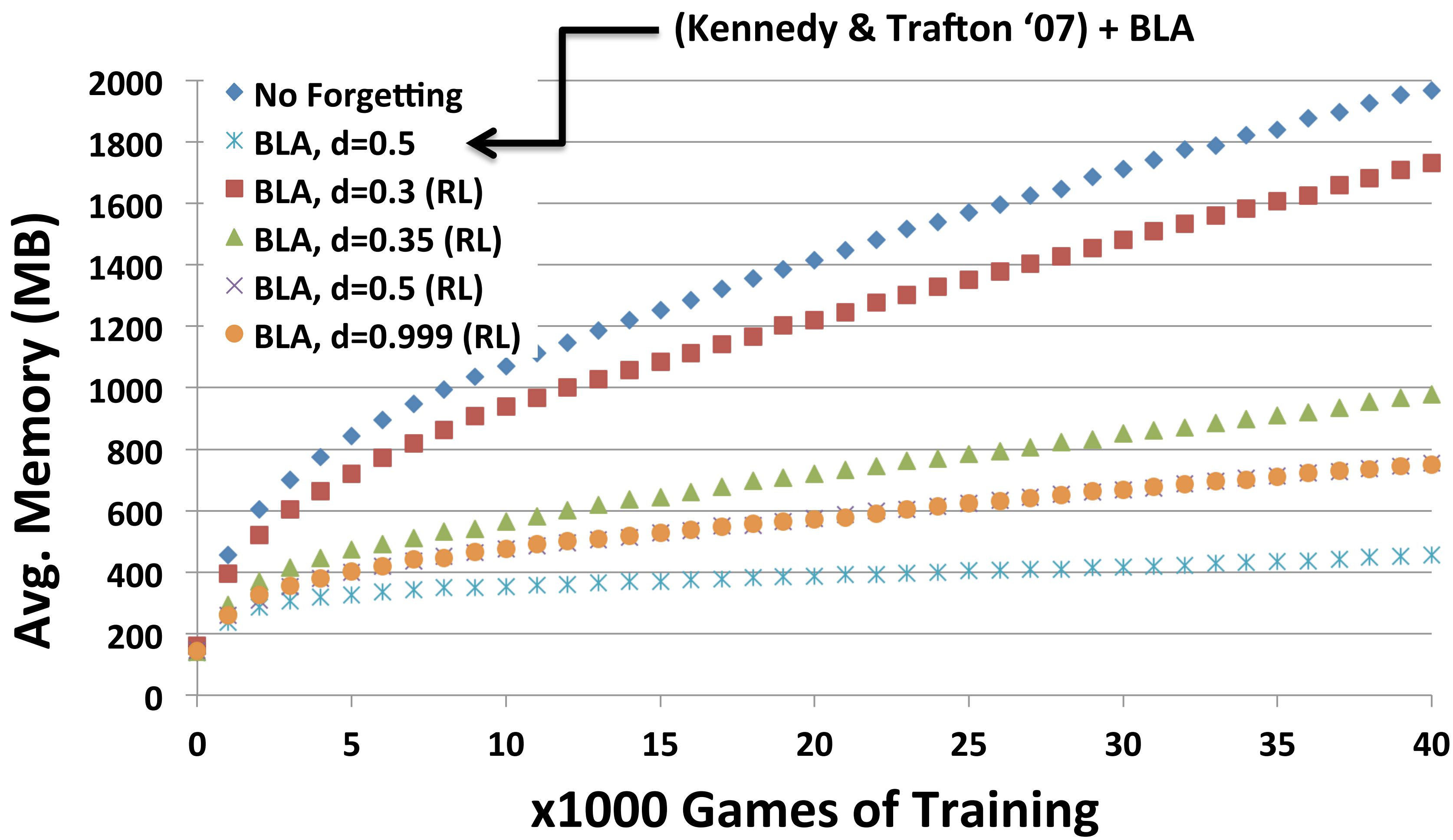
1. Forget unused RL rules that have not been rewarded
2. Learn rules via reasoning as necessary ("chunking")

Task Independent

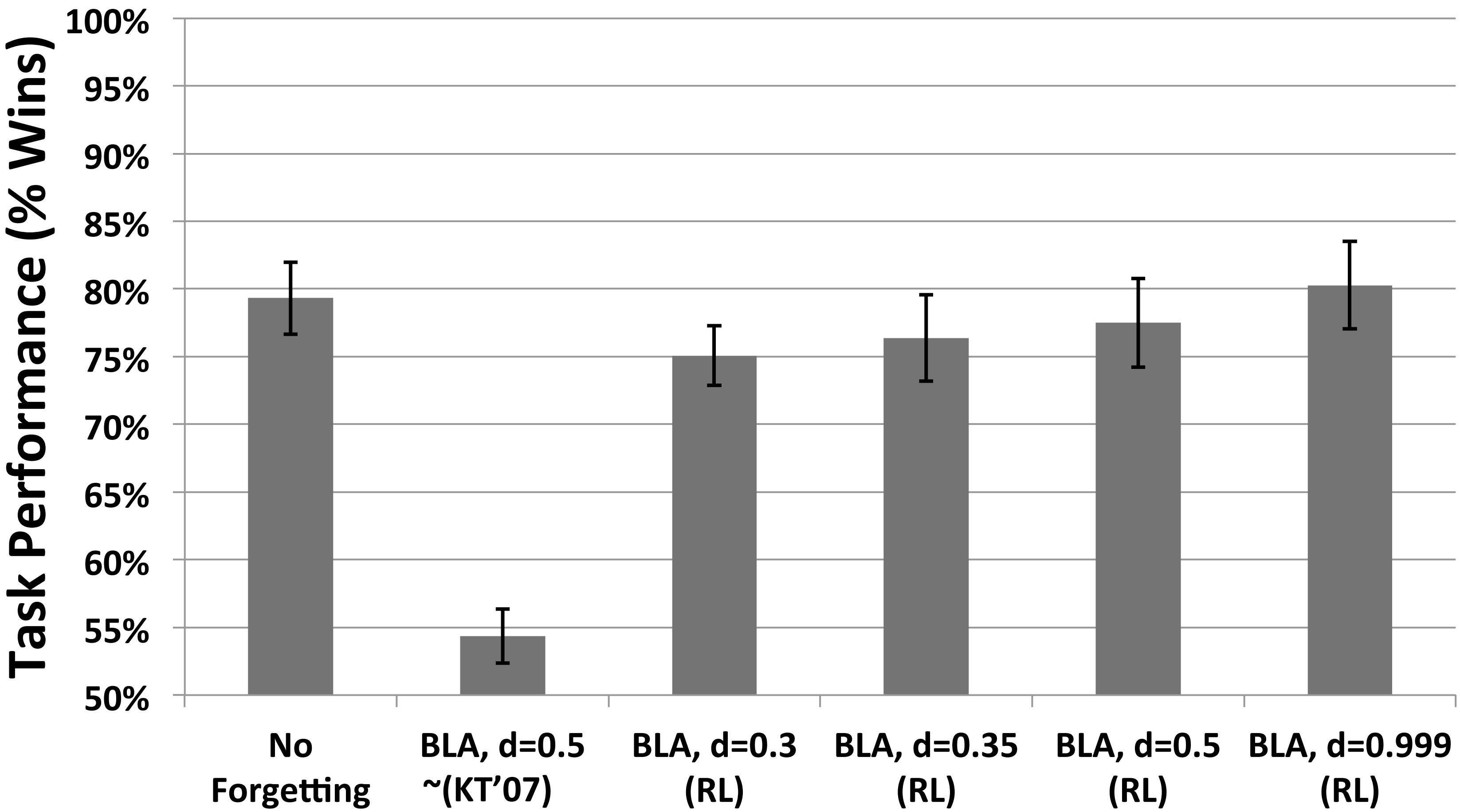
Forgetting Action Knowledge



Results: Memory Usage



Results: Competence



Summary

Explored 2 instances of common hypothesis

– Forget knowledge if not useful and can likely reconstruct if necessary

Useful for 2 long-lived models in Soar

– Bounded decision time of mobile robot (1 hour) via forgetting in working memory
– Reduced memory consumption of dice player (days) via forgetting in procedural memory

Thank You :)

Questions?