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

Nate Derbinsky, John E. Laird

University of Michigan

Motivation

Goal. Agents that exhibit human-level intelligence and persist autonomously for long periods of time (days – months).

Problem. Extended tasks that involve amassing large amounts of knowledge can lead to performance degradation.

Common Approach

Forgetting. Selectively retain learned knowledge.

Challenge. Balance…

- agent task competence &
- computational resource growth

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 in 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; Implemented in Soar v9.3.2

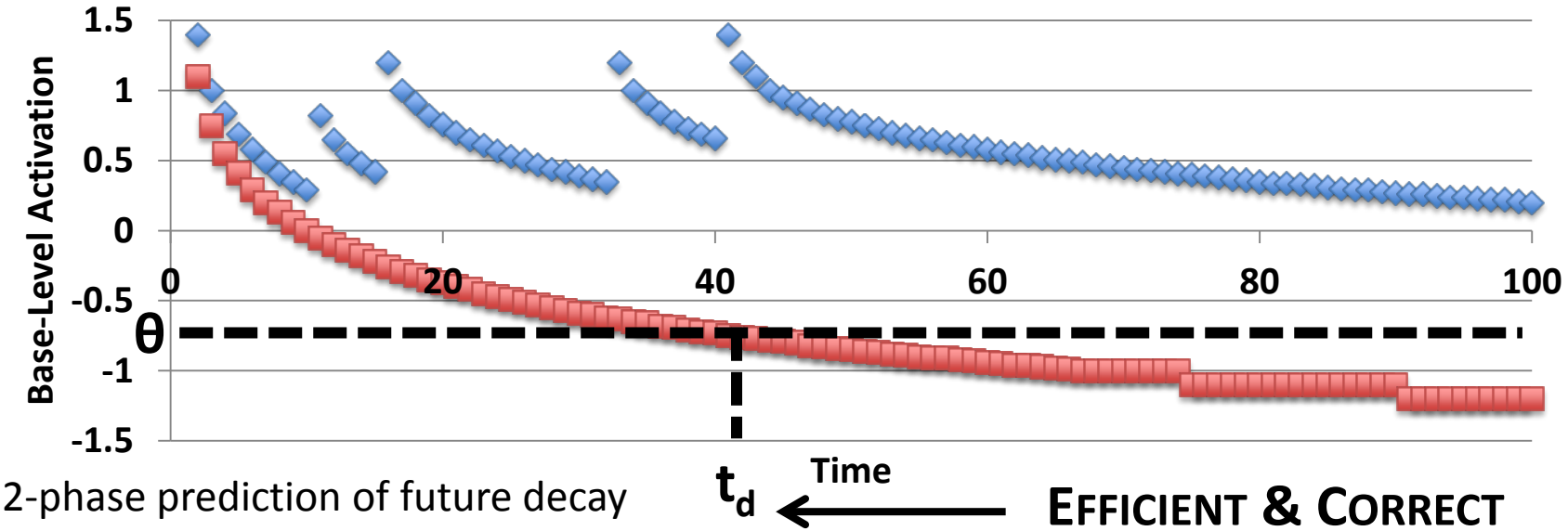
Base-Level Activation

(Anderson et al., 2004)

Predict future usage via history

- Used to bias ambiguous semantic-memory retrievals (AAAI '11)

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



2-phase prediction of future decay

- Novel approximation
- Binary parameter search

EFFICIENT & CORRECT

~~*O*(# memories)~~

Task #1: Mobile Robotics

Simulated Exploration & Patrol

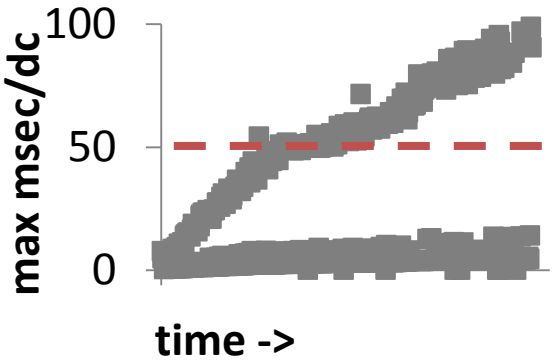
- 3rd floor, BBB Building, UM
  - 110 rooms
  - 100 doorways
- Builds map in memory from experience

Problem: Decision Time

Issue. Large working memory

- Minor: rule matching (Forgy, 1982)
- Major: episodic reconstruction

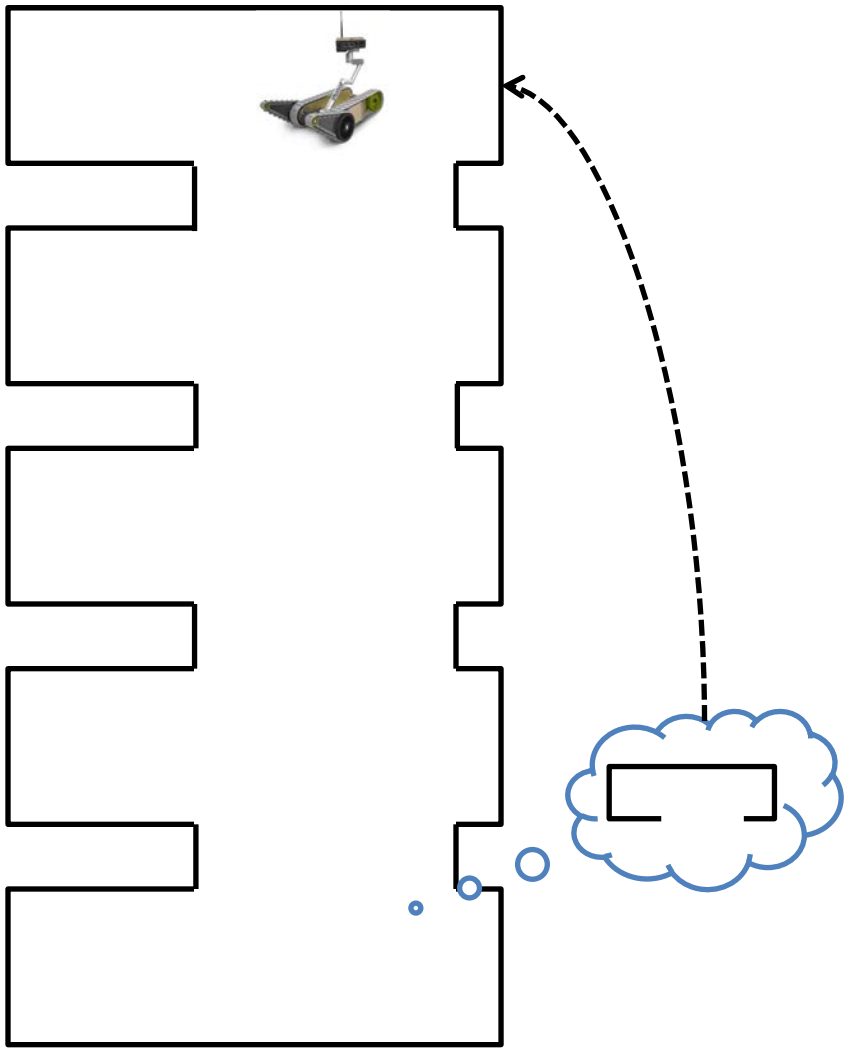
  episode size ~ working-memory size



Forgetting Policy. Memory hierarchy

1. Automatically remove from WM the o-supported augmentations of LTIs that have not been tested recently/frequently (all or nothing w.r.t. LTI)
2. Agent deliberately performs retrieve commands from SMem as necessary

Map Knowledge



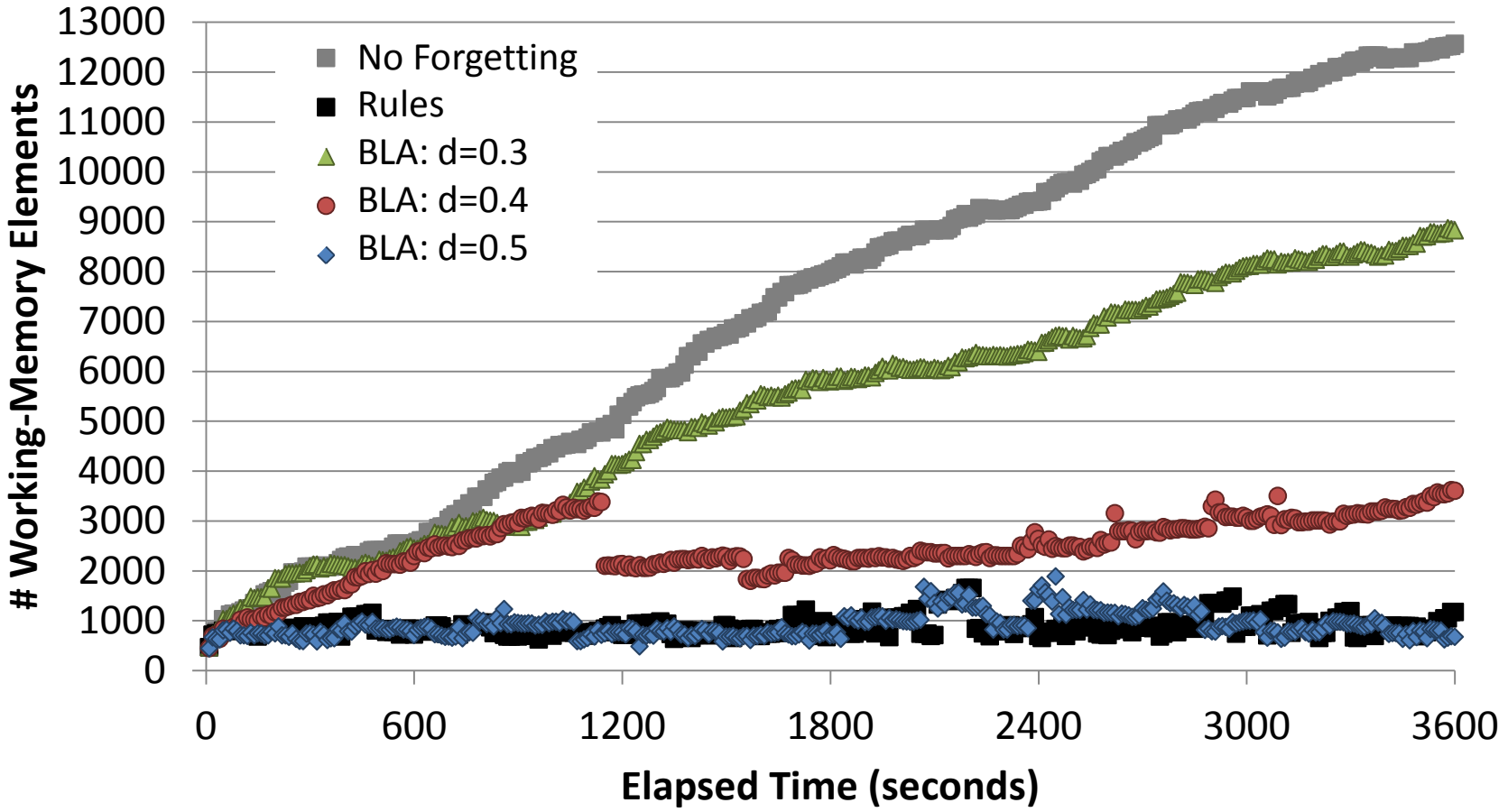
Room Features

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

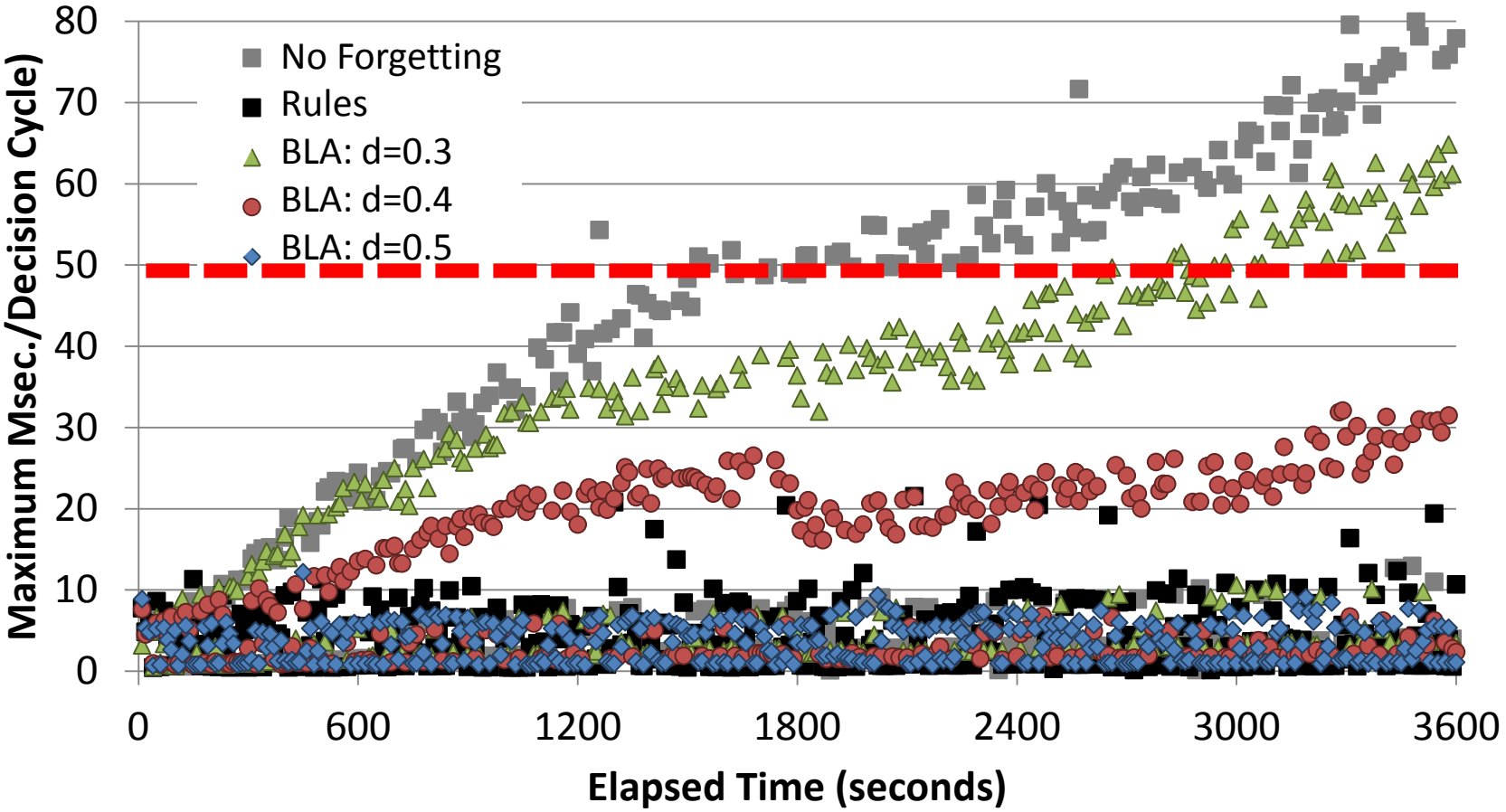
Usage

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

Results: Working-Memory Size



Results: Decision Time



Task #2: Liar's Dice

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

Reasoning --> Action Knowledge

State

Bid 6 4's

Bid 3 1's

Challenge!

1.6

0.8

-0.2

Problem: Memory Growth

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

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

Forgetting Policy. Keep what you can't reconstruct

1. Automatically excise RL chunks that have not been updated via RL and haven't fired recently/frequently
2. New chunks are learned via reasoning as necessary

Forgetting Action Knowledge

State

Bid 6 4's

1.6

Bid 3 1's

0.8

0.8

...

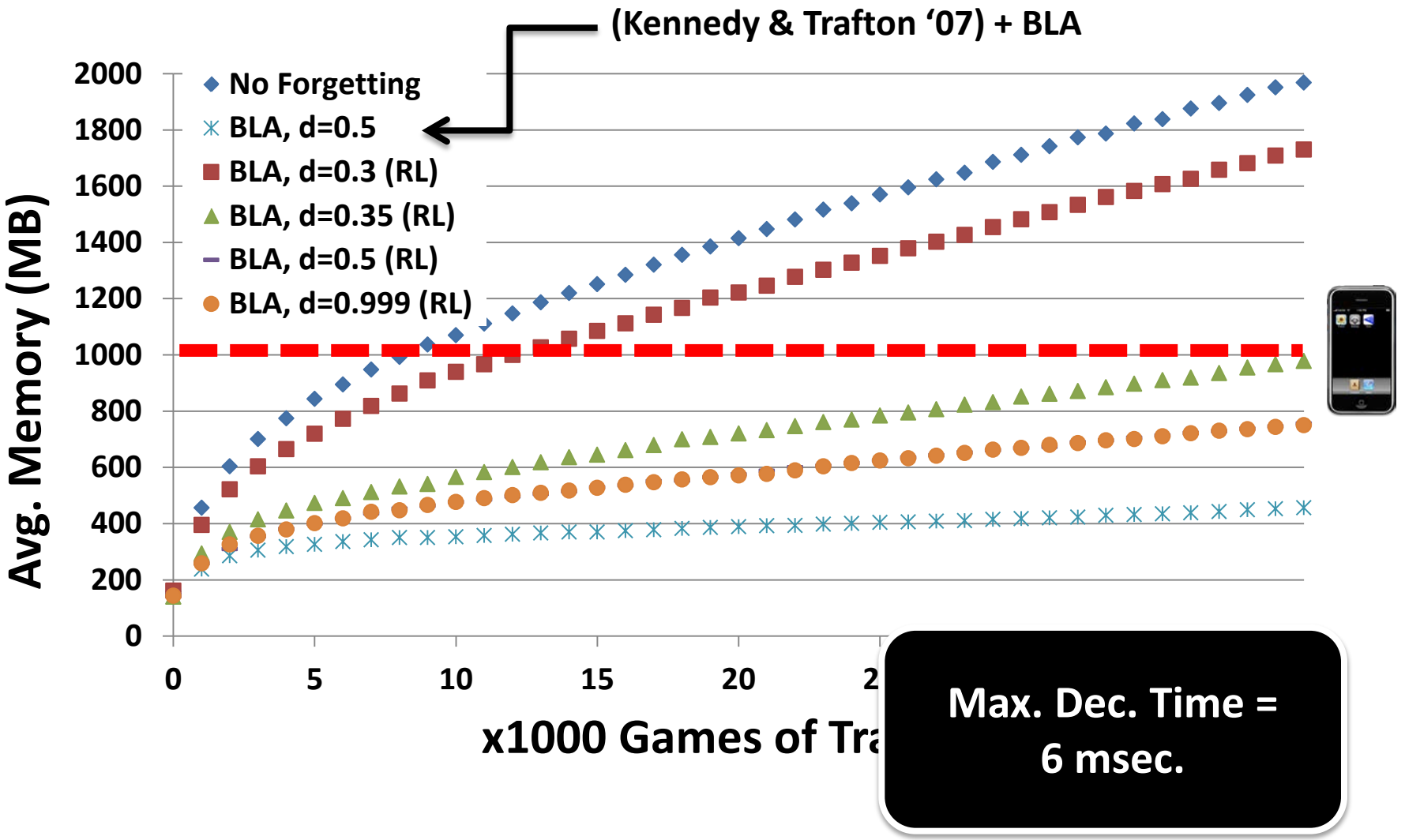
Challenge!

-0.2

-0.2

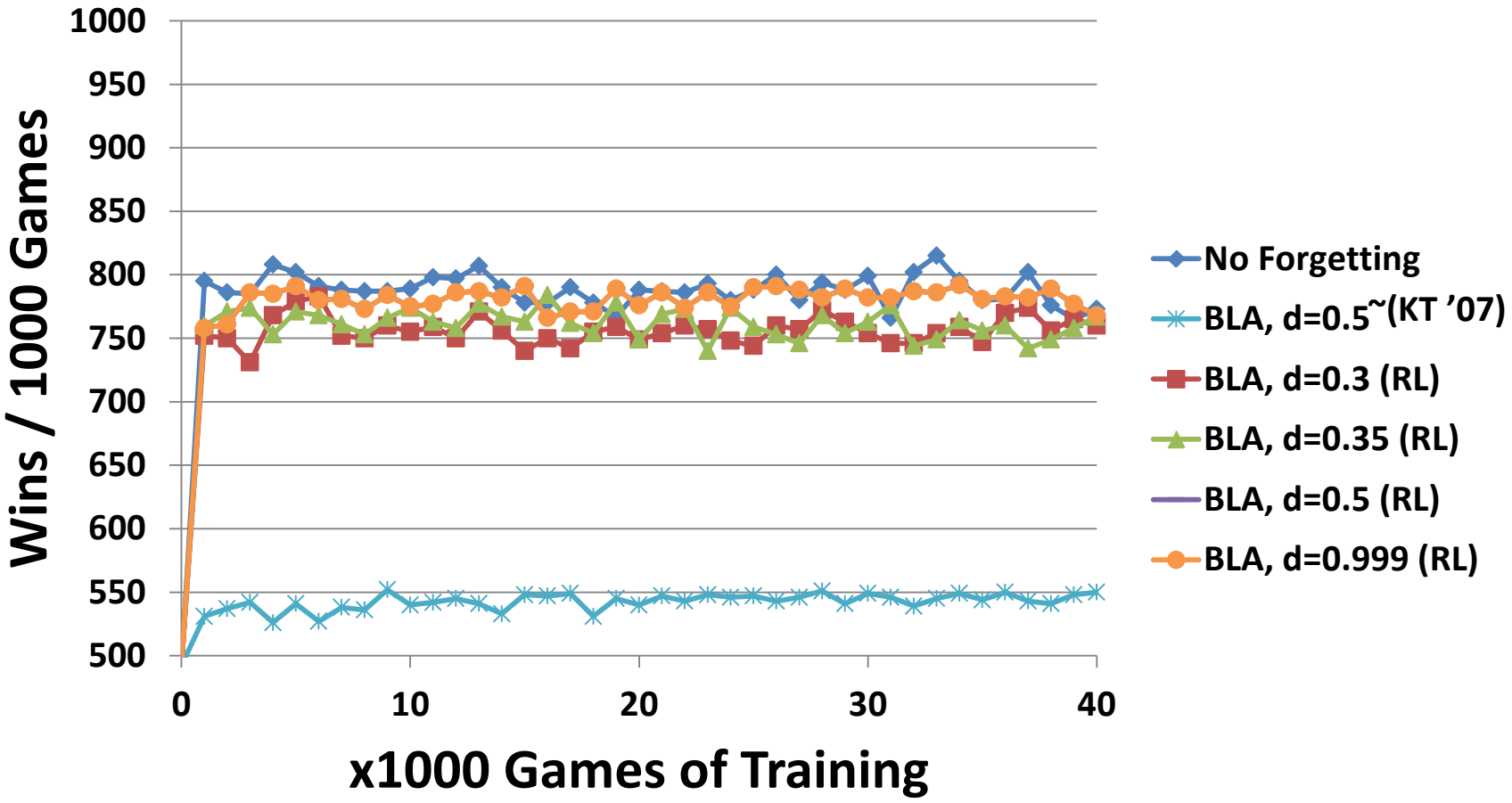
...

Results: Memory Usage



Max. Dec. Time = 6 msec.

Results: Competence



Evaluation

Nuggets

- Pragmatic forgetting policies for Soar: extends space and temporal extent of domains
  - Implemented in Soar v9.3.2
- Efficient forgetting code can be applied to any instance-based memory

Coal

- Limited domain evaluation
- Yet another set of parameters (d, θ) x 2
- **EpMem**/SMem?

For more details, see two papers in proceedings of ICCM 2012