Effective and Efficient Memory for Generally Intelligent Agents

Nate Derbinsky

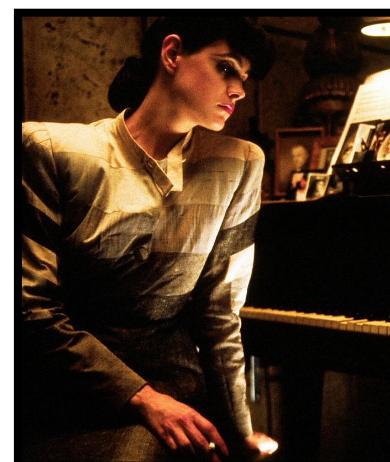
This Work: *Soar Group, University of Michigan*
Advisor: John Laird

Present: *PostDoc @ Disney Research*
with: Jonathan Yedidia

Long-Term Research Goal

General Intelligence

Agents that persist for long periods of time, exhibiting robust and adaptive behavior in a variety of tasks and situations.



Inspiration from Humans: Memory

Class of mechanism to cope with dynamic, partially-observable environment

- **Encodes** experience
- **Stores** internally
- Supports **retrieval**

Without memory, agents are reactive, stuck in the *here* and *now*.

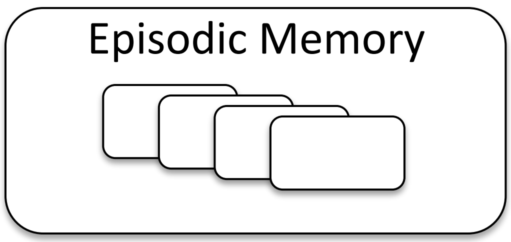
Computational Challenge of Memory

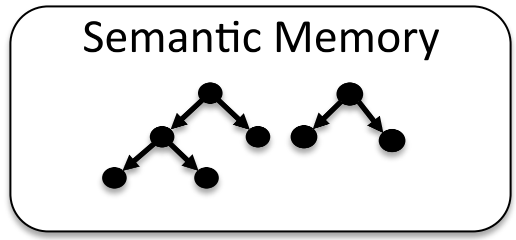
How to maintain effective and efficient access to large amounts of knowledge as it accumulates over long periods of time.

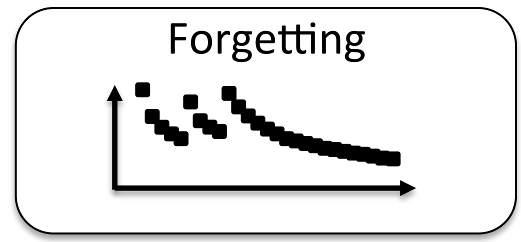
Limitations of prior approaches…

- task-specific,
  (*e.g. Macedo & Cardoso, 2004*)
- restricted representation, and/or
  (*e.g. Tecuci & Porter, 2007; 2009*)
- do not scale to large amounts of experience
  (*e.g. Kuppuswamy et al., 2006; Douglass et al., 2009*)

Effective and Efficient Memory





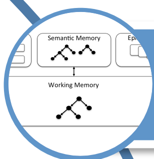


Desiderata

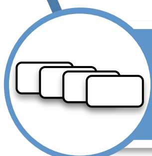
- **Generality**: effective across a variety of tasks
- **Reactivity**: decisions < 50 milliseconds
- **Scalability**: support large amounts of knowledge

NOT comparing to human memory/data!

Outline



Cognitive Architecture & Soar

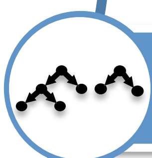


Episodic Memory (overview)

[ICCBR '09], [BRIMS '11], [AAMAS '12], [AAAI '12a], [AAAI '12b]

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

Base-Level Decay



Semantic Memory

[ICCM '10], [BRIMS '11], [AAAI '11], [AAAI '12b], [ICCM '12a]

Forgetting

[ICCM '12a], [ICCM '12b], [ACS '12], [CSR '13]

Cognitive Architecture
(Newell, 1990)

Specification of those aspects of cognition that remain constant across the lifetime of an agent

- Memory systems of agent's beliefs, goals, experience
- Knowledge representation
- Functional processes that lead to behavior
- Learning mechanisms

Goal. Develop and understand intelligence across a diverse set of tasks and domains

Cognitive Architectures
Commonalities & Differences

Theory

- Knowledge representation
- Processes (e.g. decision-making, action, learning)

Methodology

- Research focus/evaluation criteria

Practicality

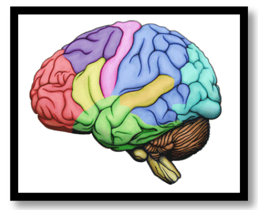
- Hardware/software platforms
- Implementation reliability & support
- Reactivity & scalability

Research Focus

Biological Plausibility

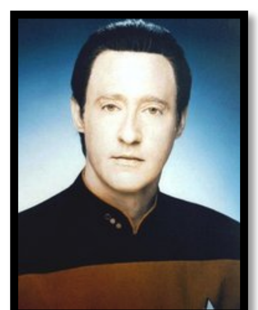
Leabra

Psychological Plausibility



ACT-R
CLARION
EPIC

Agent Functionality

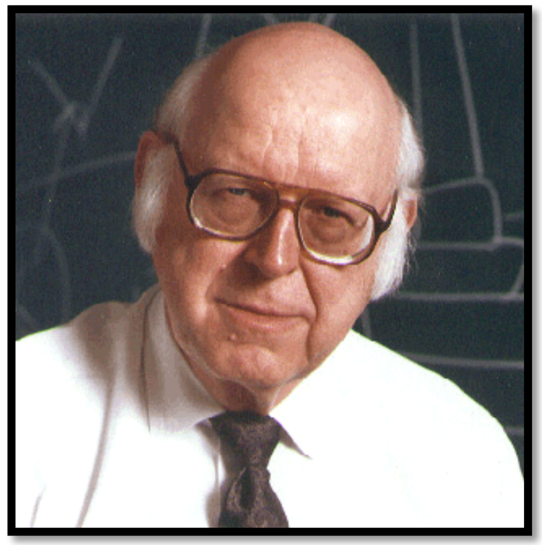


Companions
ICARUS
LIDA
Sigma
Soar

The Soar Cognitive Architecture

Created in 1982 by…

John Laird
Professor
Michigan



Allen Newell
Founder of AI

Paul Rosenbloom
Professor
USC, ICT

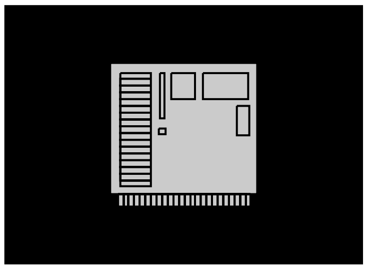
Soar

Distinctive Characteristics

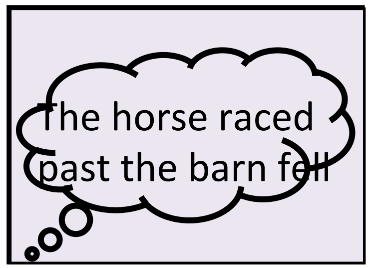
- Efficiently brings to bear large amounts of knowledge
- Diverse mechanisms that support general problem solving methods
- Public distribution and documentation
  - Major operating systems (Windows, OS X, Linux)
  - Many languages (C++, Java, Python, …)

- Annual Soar Workshop
  - Free @ UM, Ann Arbor: June 3-4 (tutorials), June 5-7 (talks)
  - Academic, Government, Corporate (incl. SoarTech)

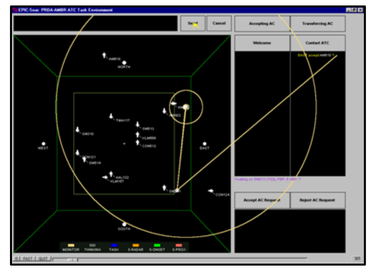
Soar
Select Applications (1)



R1-Soar
Computer Configuration

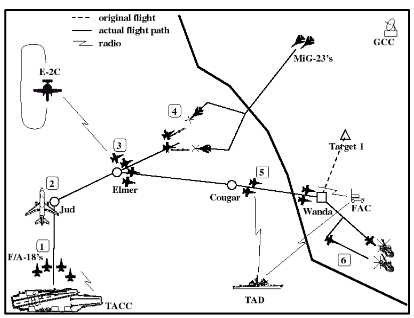


NL-Soar
Language Processing

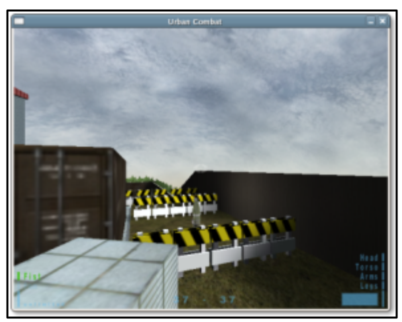


Amber EPIC-Soar
Modeling HCI

ICT Virtual Human
Natural Interaction, Emotion



TacAir-Soar
Complex Doctrine & Tactics



Urban Combat
Transfer Learning

Soar Quakebot
Anticipation

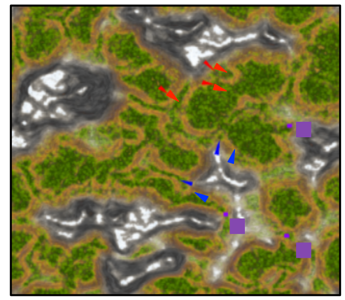
Haunt
Actors and Director

Soar

Select Applications (2)

MOUTbot
Team Tactics &
Unpredictable Behavior

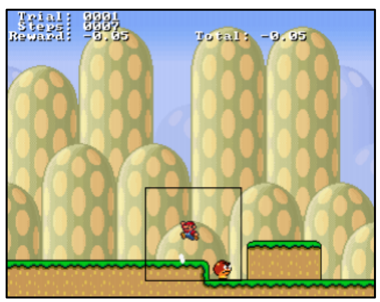
SORTS
Spatial Reasoning &
Real-time Strategy



Simulated Scout
Mental Imagery

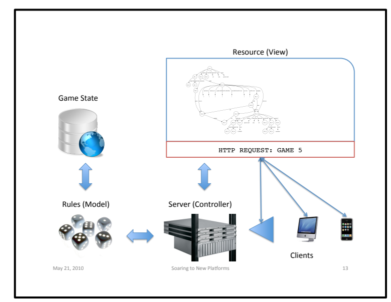
Splinter-Soar
Robot Control

ReLAI
Mental Imagery &
Reinforcement Learning



Infinite Mario
Hierarchical
Reinforcement Learning

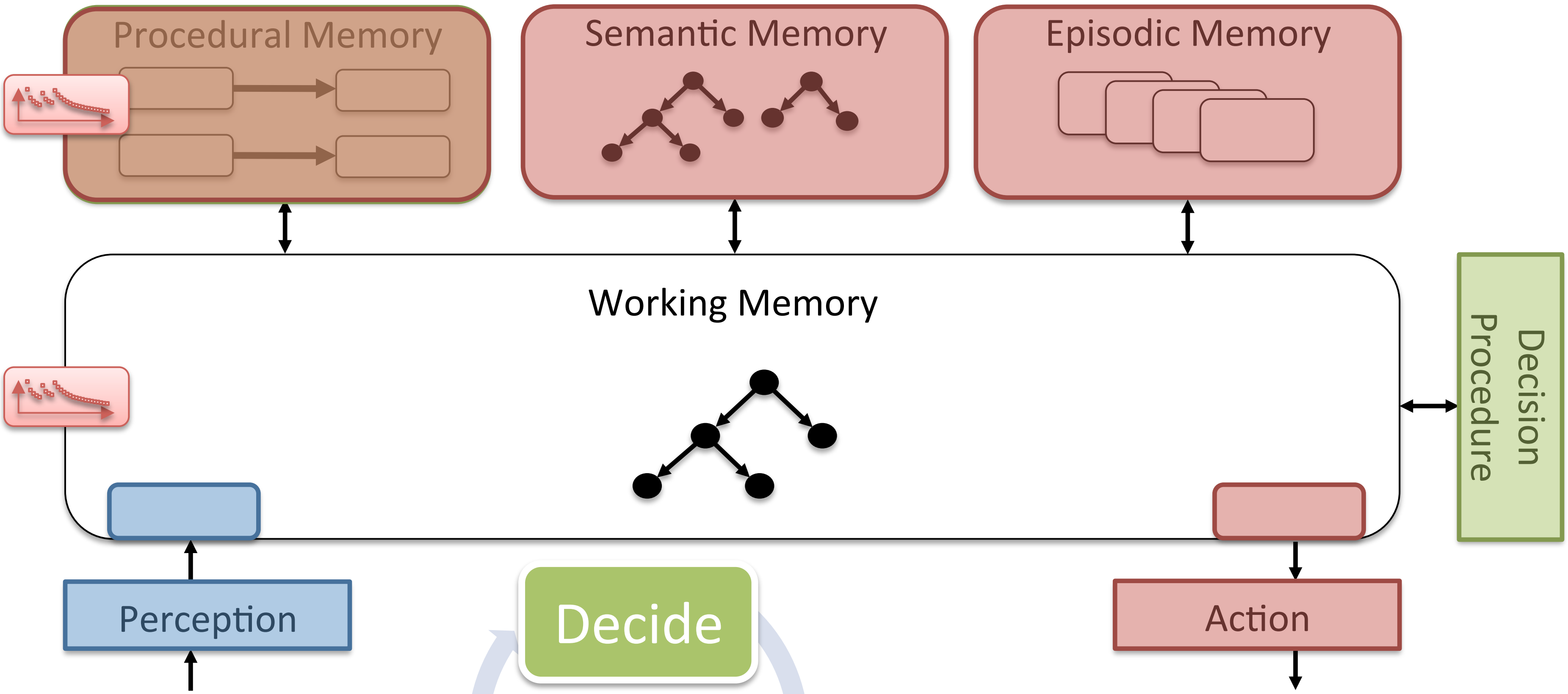
iSoar
Mobile Reinforcement
Learning

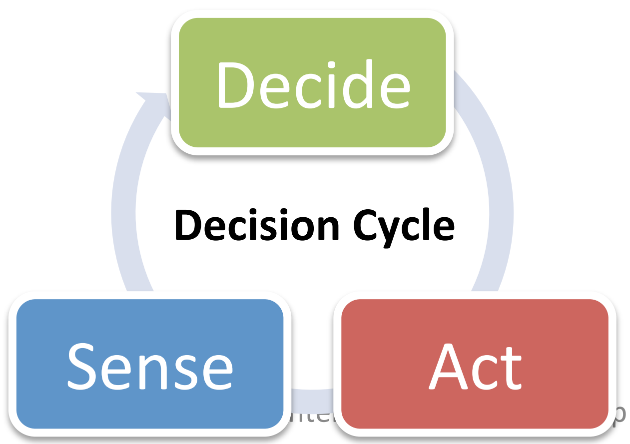


RESTful Soar
Web-based Gameplay,
Probabilistic Learning

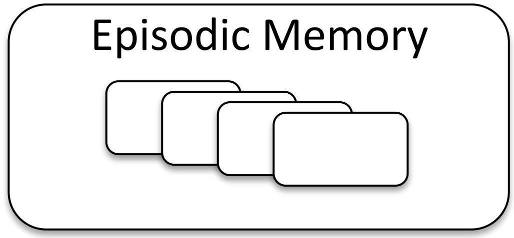
Soar (Laird, 2012)

Memory Integration





Agent **reactivity** is the time required to execute each decision cycle, including LT memory access ∴ only bounded search!



Long-term, contextualized store of specific events (Tulving, 1983)

Episodic Memory

Integration

Representation

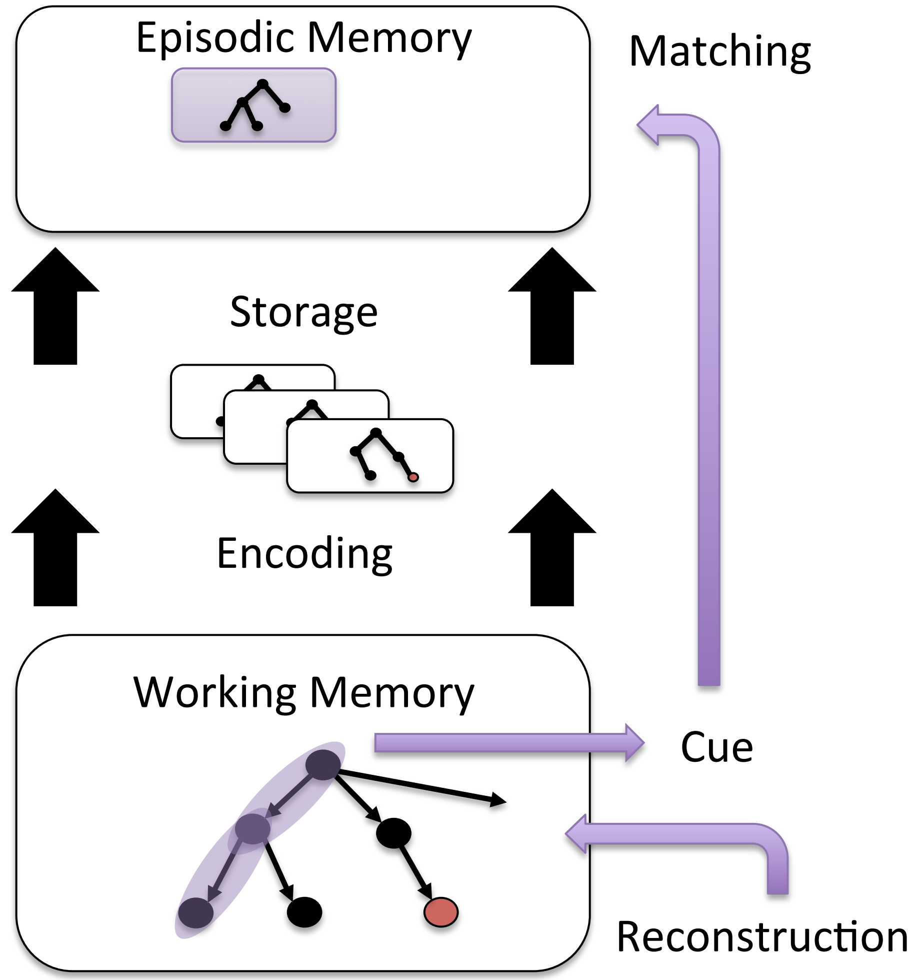
- Episode: connected di-graph
- Store: temporal sequence

Encoding/Storage

- Automatic
- No dynamics (e.g. forgetting, blending, …)

Retrieval

- Cue: acyclic graph
- Semantics: desired features in context
- Find the most recent episode that shares the most leaf nodes in common with the cue



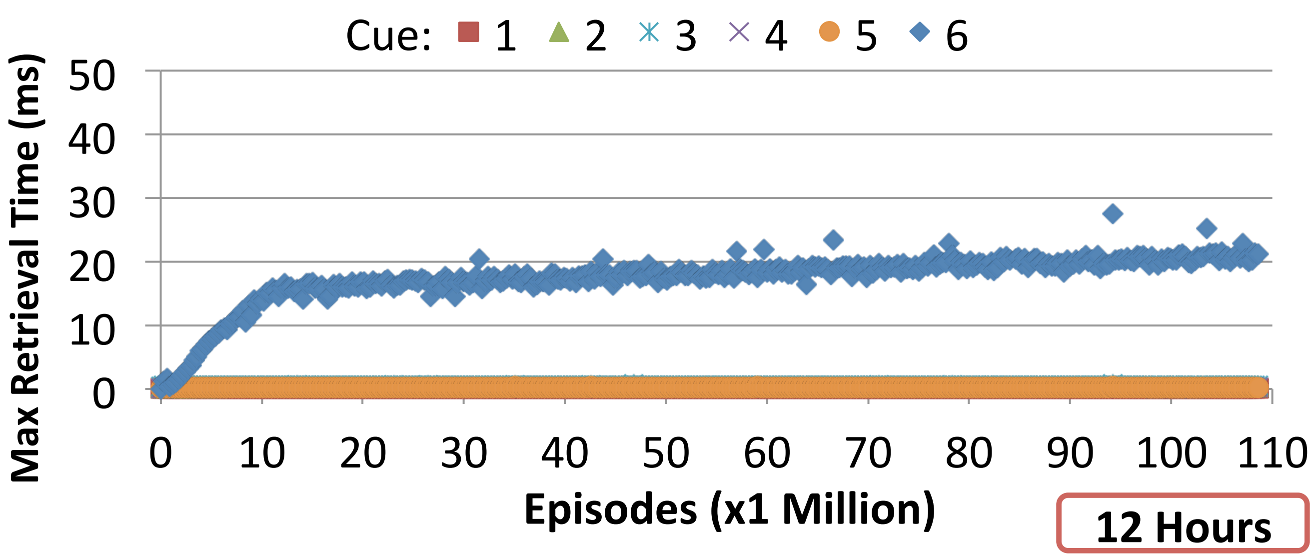
Empirical Evaluation

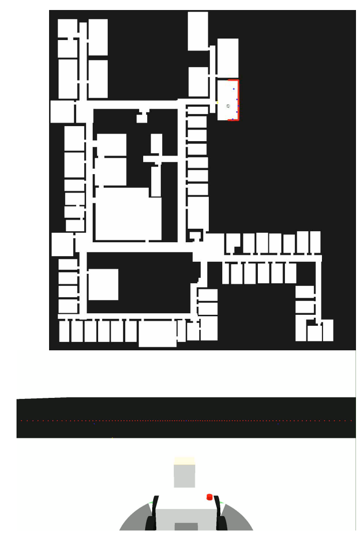
Analysis & Algorithms 

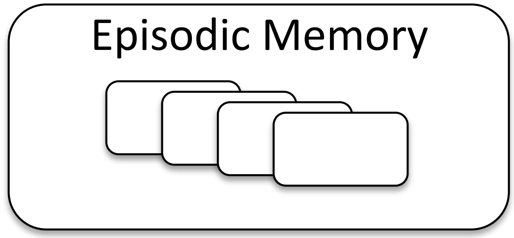
- *Please ask during Q&A, offline, etc.*

Experimental Setup 

- 49 domains: WSD, planning, robotics, games
- 10^5-10^8 episodes ~ days of real time, >100 cues







- Algorithms that are reactive and scalable for many tasks and cues
- Performance characterization w.r.t. general properties of environments, tasks, and agents
- Demonstrated useful capabilities in a variety of problem domains

Ongoing Research

- Learning to use memory (Gorski '12)
- Prospective memory (Li et al. '12)
- Mixed-initiative situated instruction (Mohan et al. '13)
- Bounding memory
- Consolidation

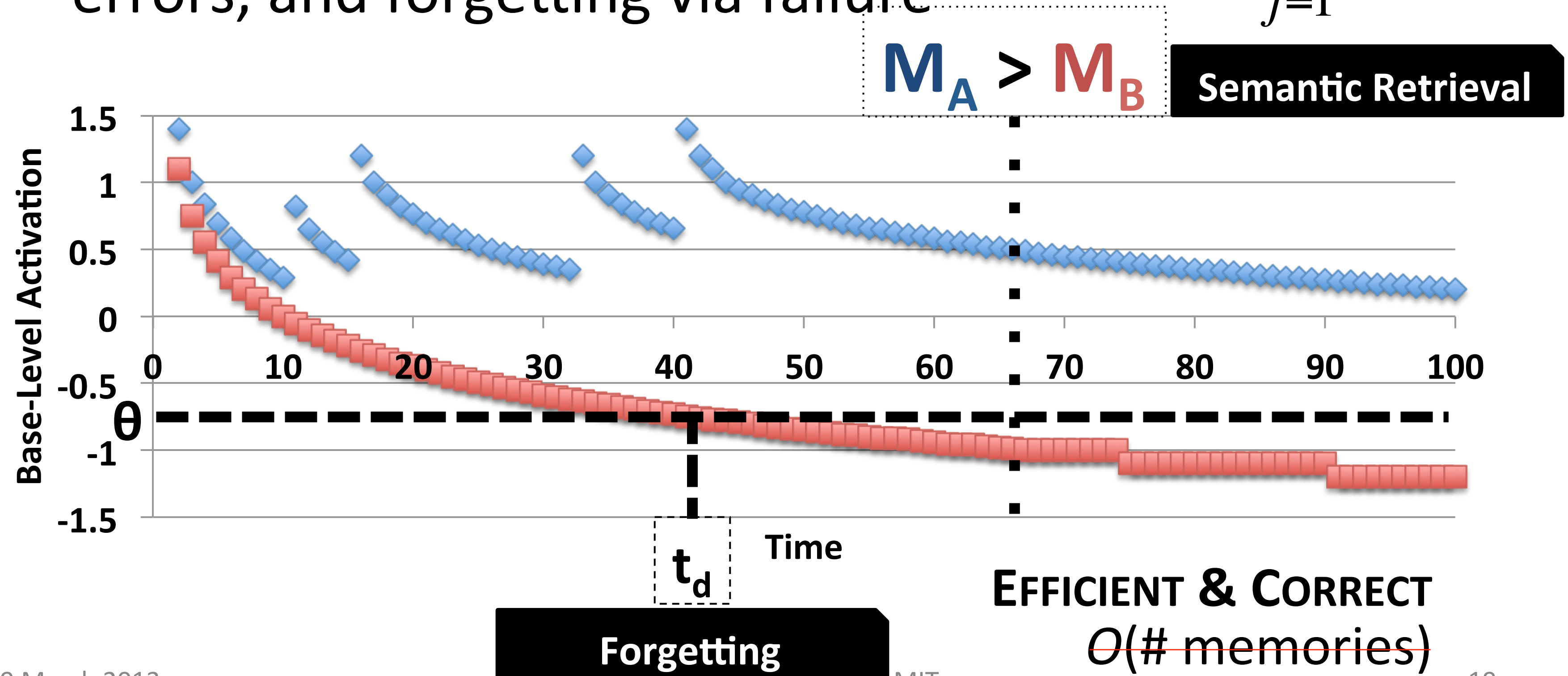
…

Base-Level Decay
(Anderson et al. 2004)

Predict future usage via history

Used to model human retrieval bias, errors, and forgetting via failure

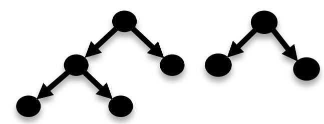
$$\ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$



EFFICIENT & CORRECT

~~*O*(# memories)~~

Semantic Memory



Long-term store of general facts and relations about the world, independent of the context in which they were originally learned

Agent Benefits

- Access to large KBs
- Retrieval bias as a reasoning heuristic

SUMO (upper ontology)

- 4.5K classes, 250K facts

WordNet (lexicon)

- 212K senses, 820K assertions

Cyc (“common sense”)

- 500K concepts, 5M facts

Semantic Memory

Integration

Representation

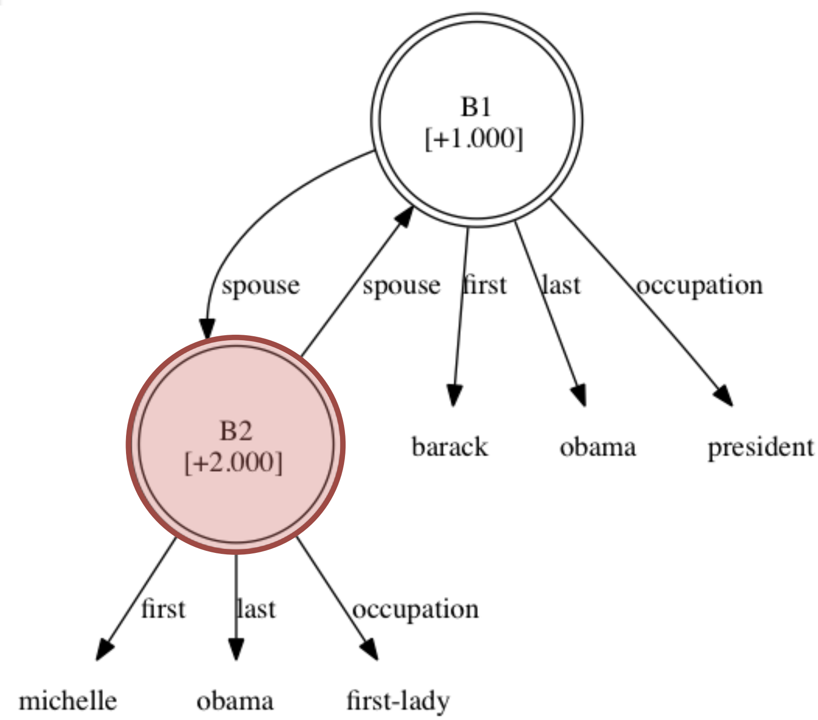
- Directed graph

Encoding/Storage

- Incremental
- Deliberate

Retrieval

- Cue: set of features/relations
- Semantics: subset query
- Single result, ranked by bias value [#]



Example cue:
`last(obama),spouse(X)`

Semantic Memory
Computational Challenges

Dynamic…

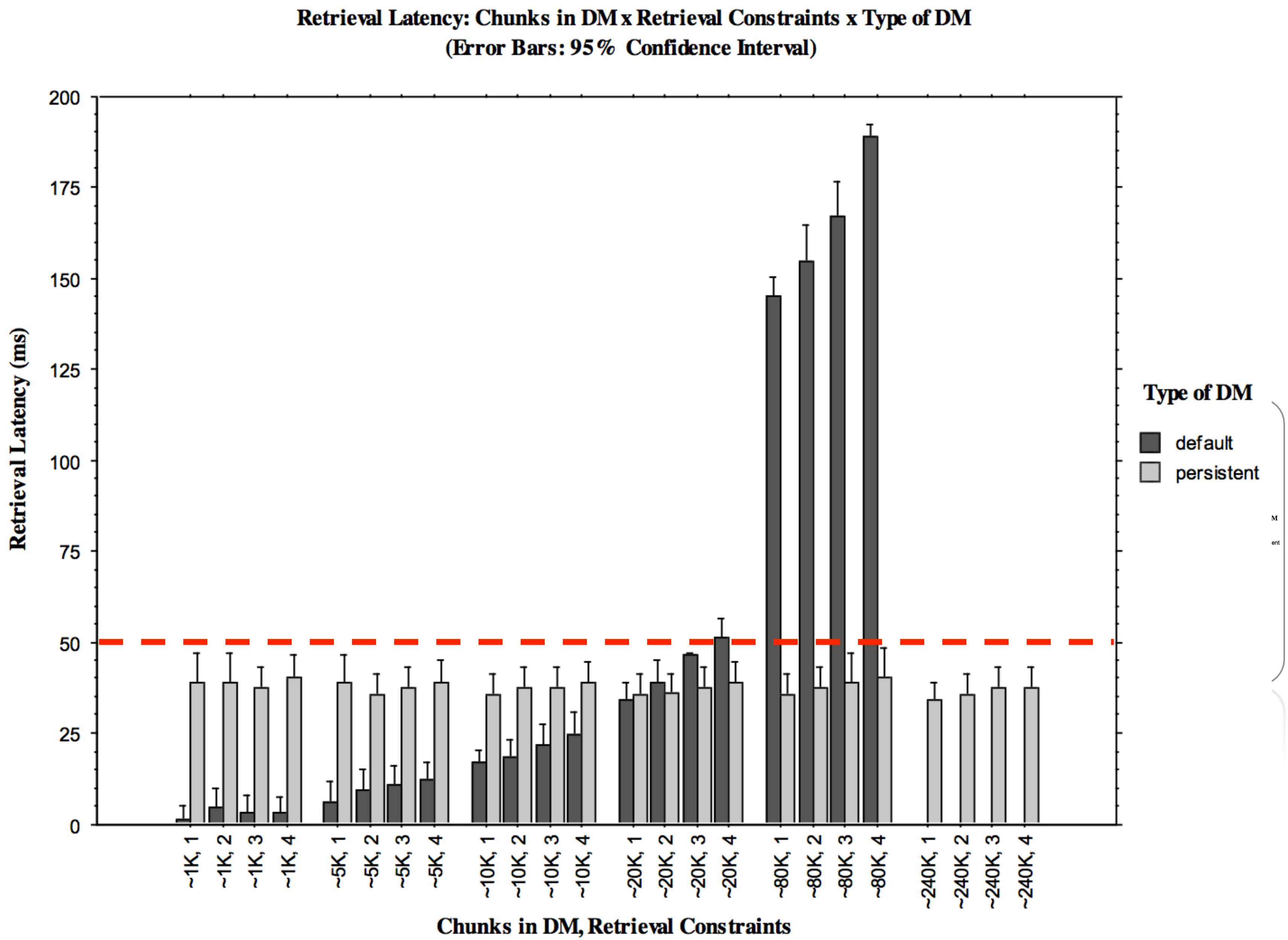
- number of nodes/edges
- symbol vocabulary

Scaling potential

- Nodes ~ millions
- Edges ~ 10 per node

Cue-matching optimality

- Feature satisfaction, ranking w.r.t. bias value
- O(|cue| x |objects|)



Retrieval Latency: Chunks in DM x Retrieval Constraints x Type of DM
(Error Bars: 95% Confidence Interval)
Retrieval Latency (ms)
200
175
150
125
100
75
50
25
0
~1K, 1
~1K, 2
~1K, 3
~1K, 4
~5K, 1
~5K, 2
~5K, 3
~5K, 4
~10K, 1
~10K, 2
~10K, 3
~10K, 4
~20K, 1
~20K, 2
~20K, 3
~20K, 4
~80K, 1
~80K, 2
~80K, 3
~80K, 4
~240K, 1
~240K, 2
~240K, 3
~240K, 4
Chunks in DM, Retrieval Constraints
Type of DM
default
persistent

Analysis & Algorithms

Storage

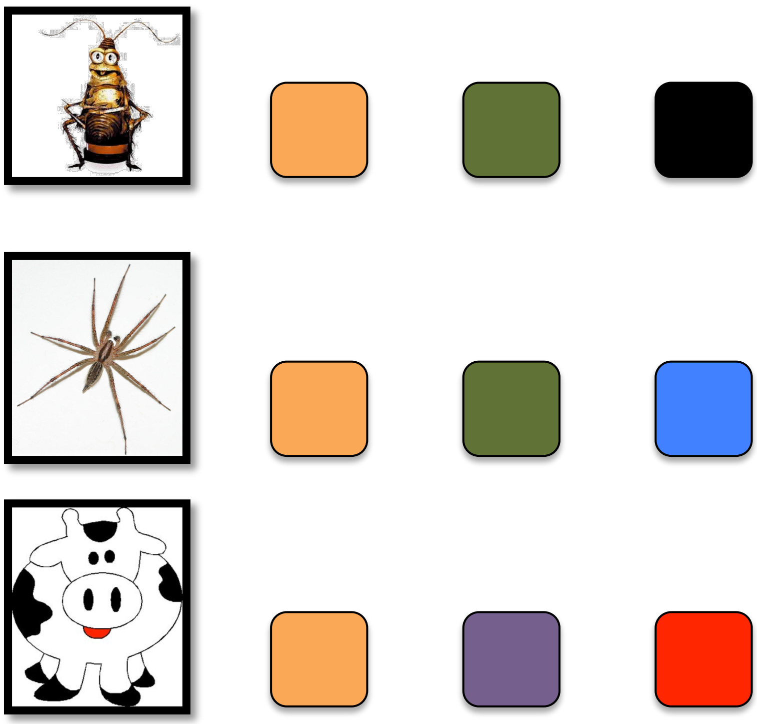
- Incremental inverted index (via b+-trees)
  (Zobel and Moffat, 2006)

Cue Matching

- Statistical query optimization (Chaudhuri, 1998)
- Hybrid ranking via *locally efficient* bias functions*

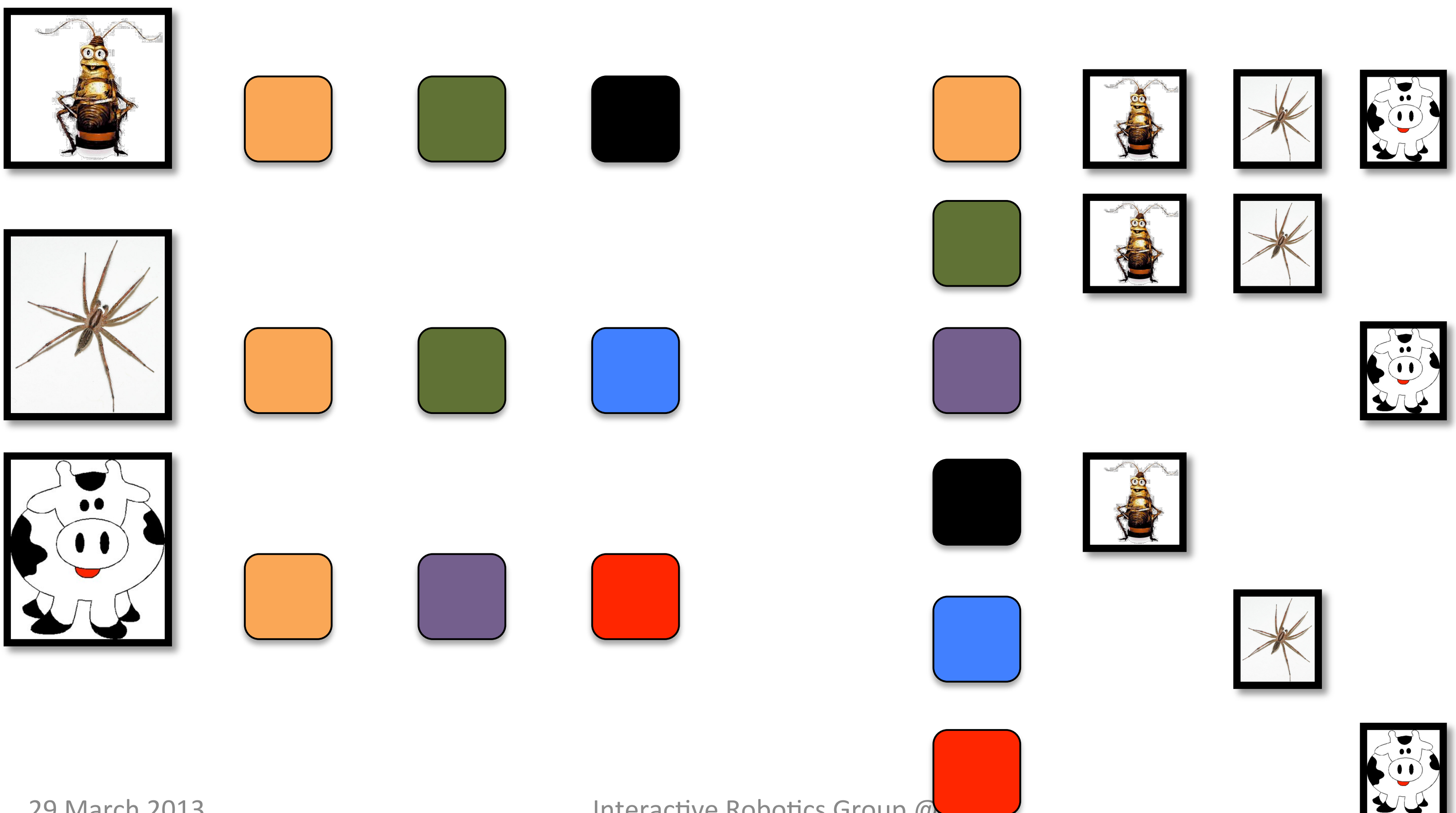
Example Semantic Knowledge

Semantic Objects: Features



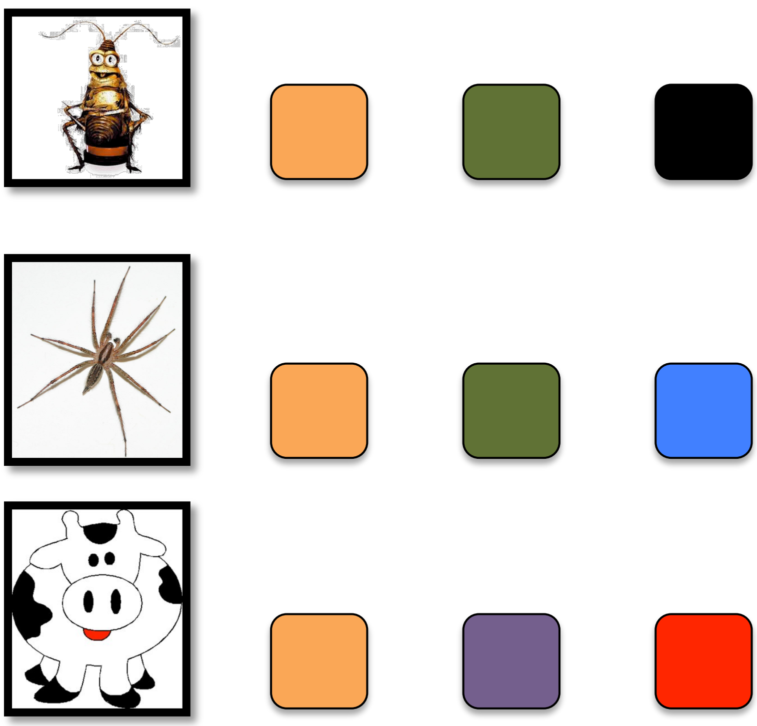
Inverted Indexing

Semantic Objects: Features **Inverted Index**

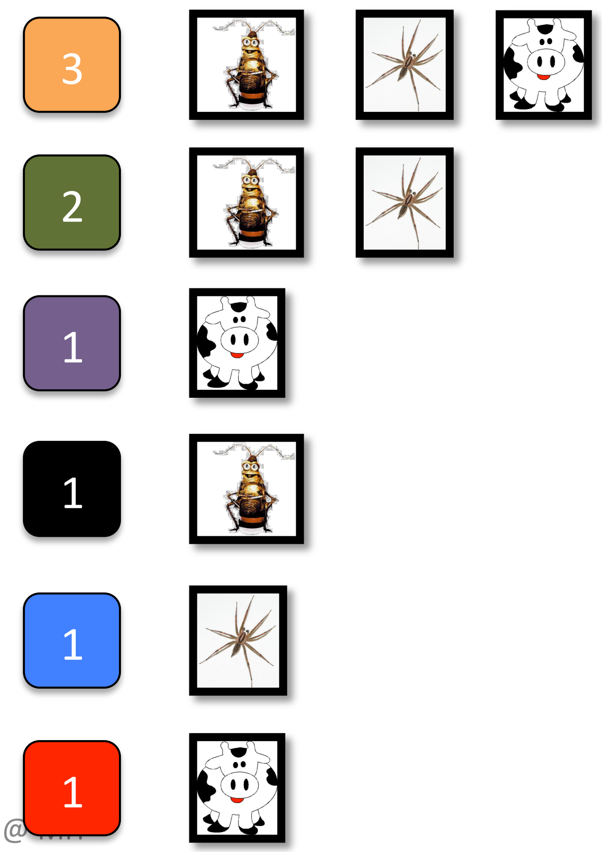


Feature Statistics

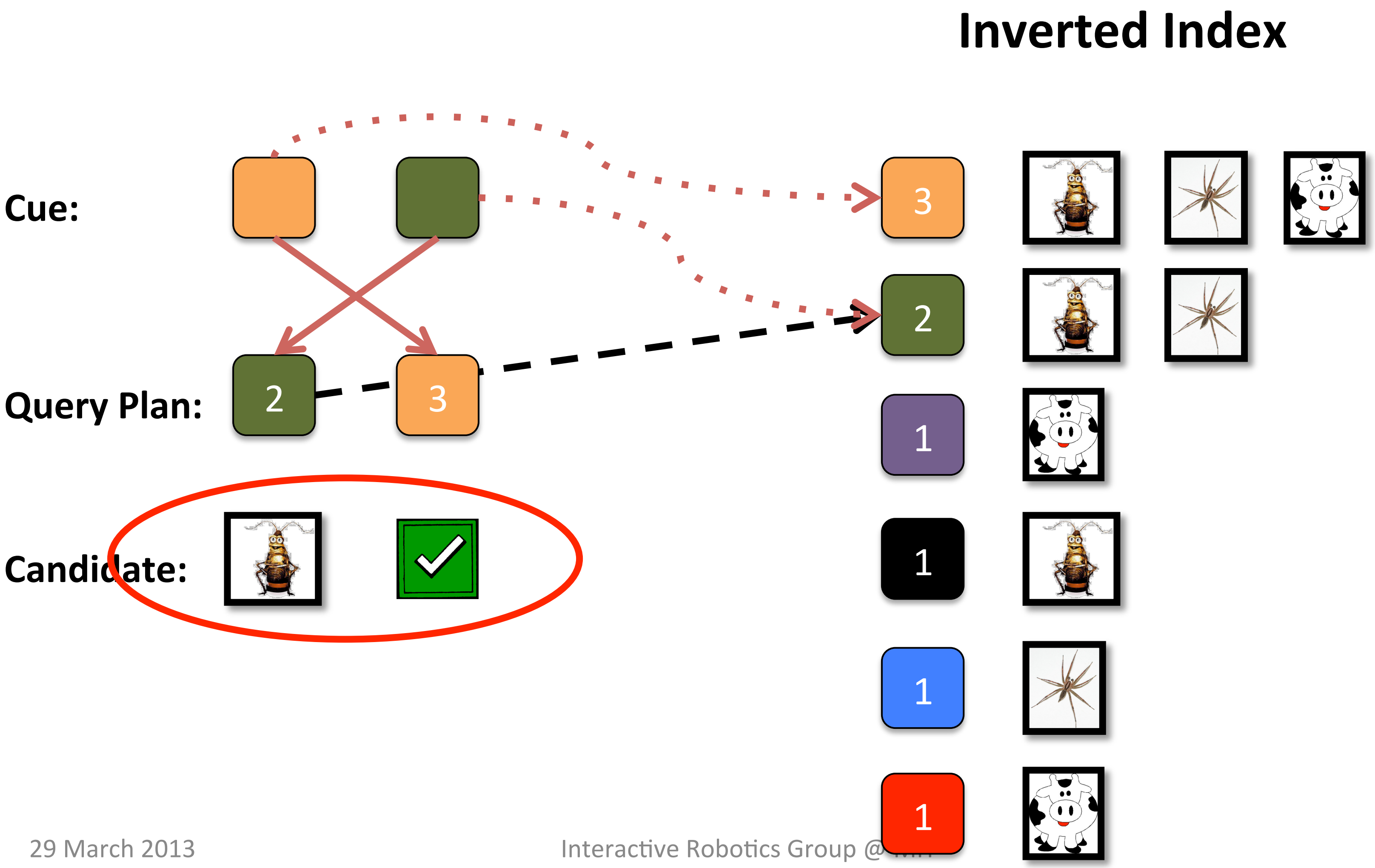
Semantic Objects: Features



Inverted Index

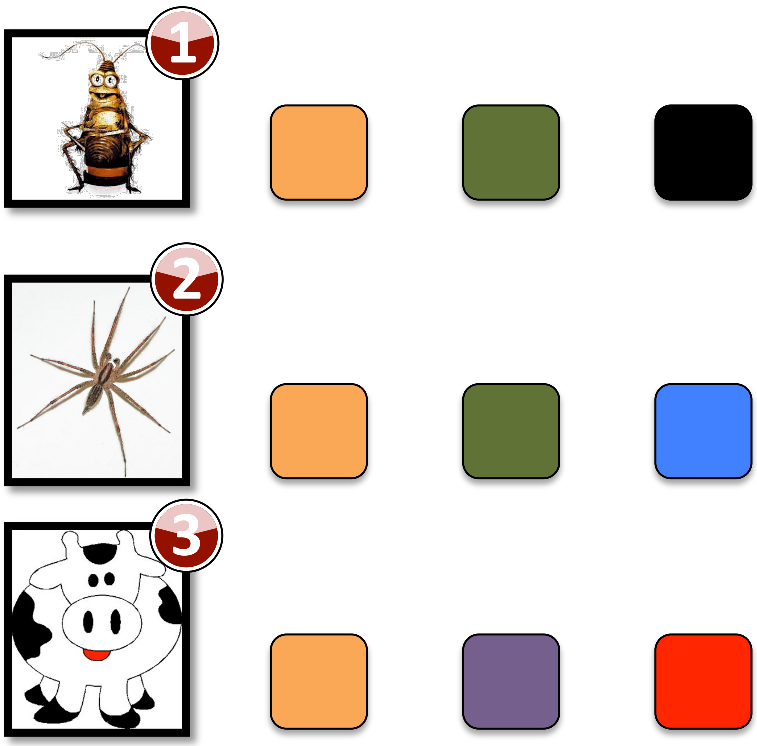


Non-Biased Retrieval Algorithm

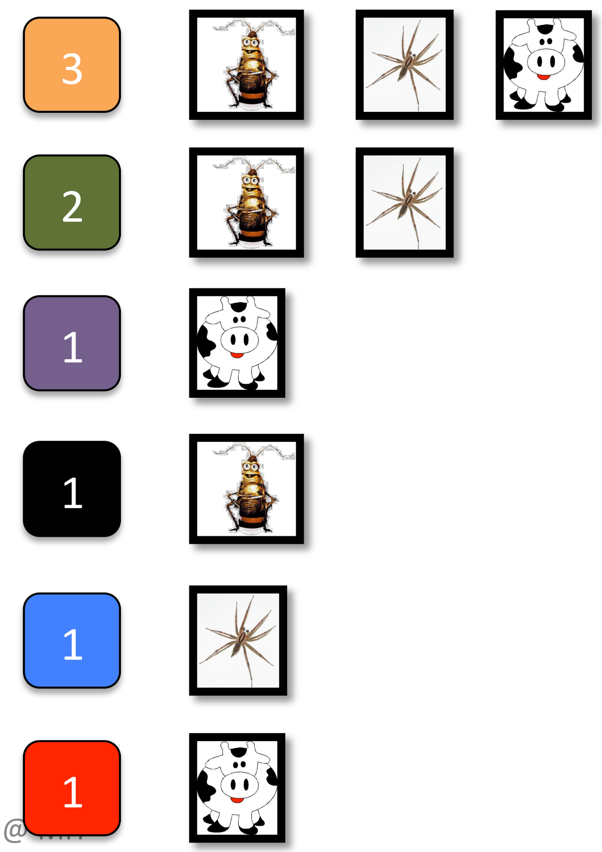


Introducing Bias

Semantic Objects: Features

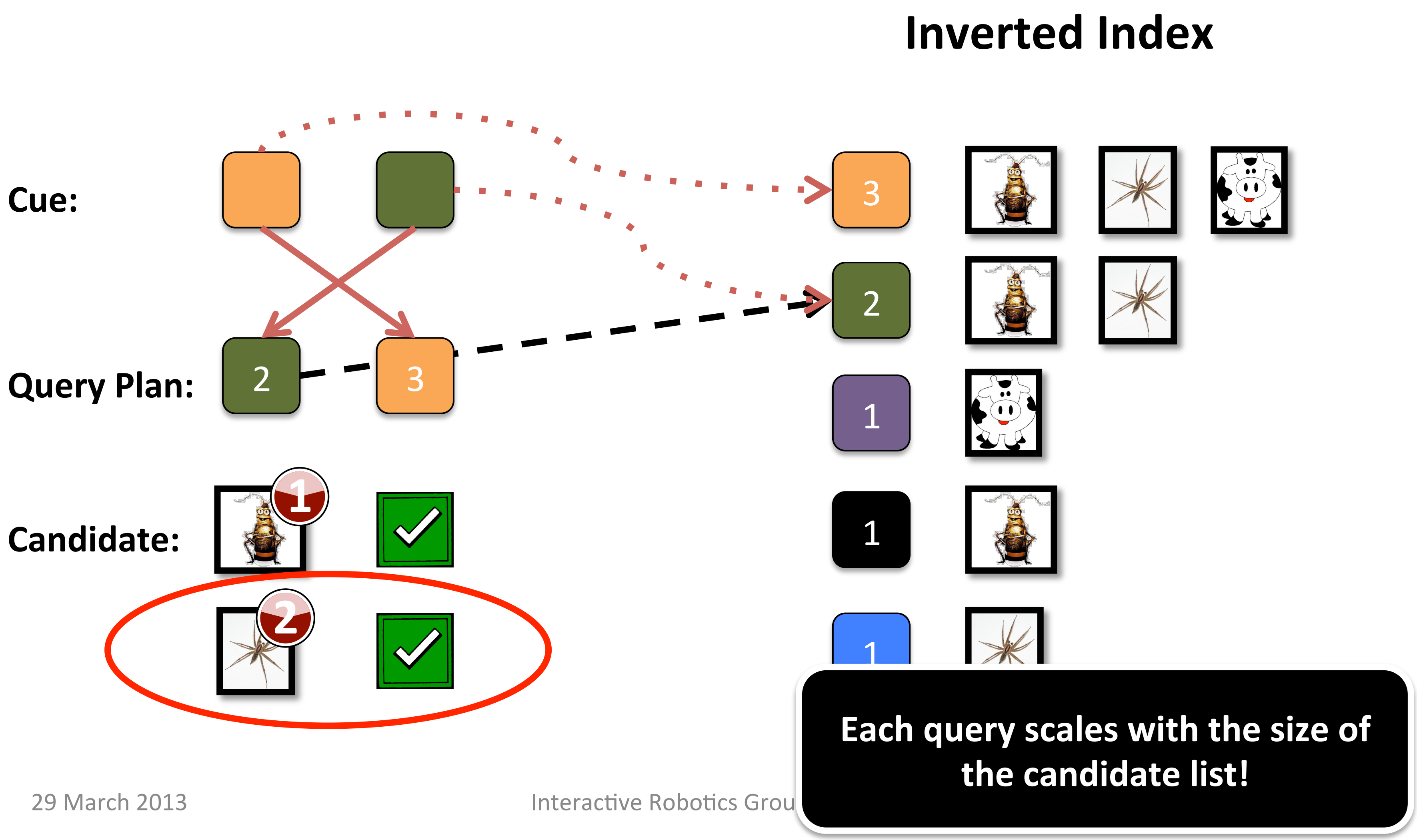


Inverted Index



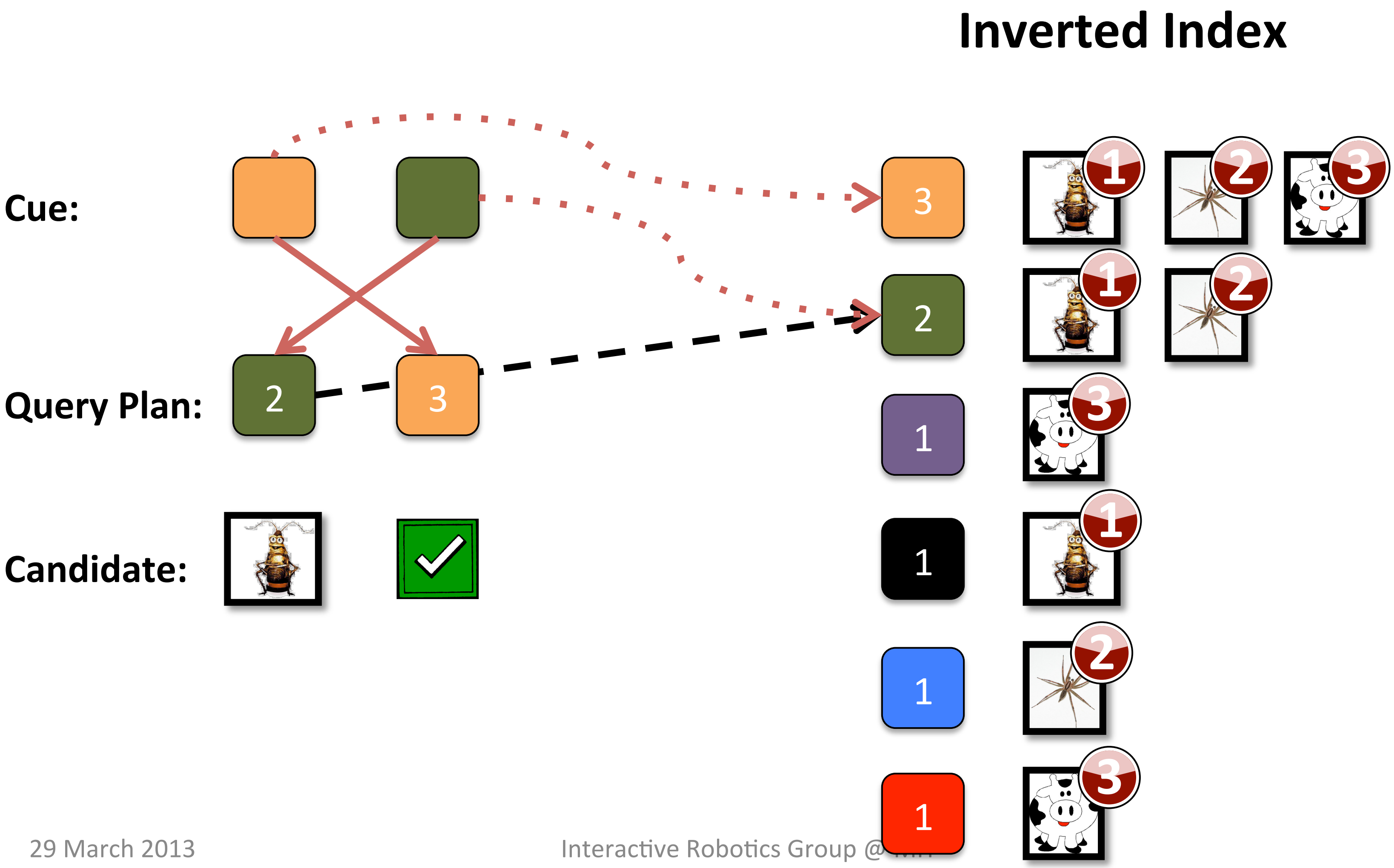
Biased Retrieval Algorithm #1

Sort on Query



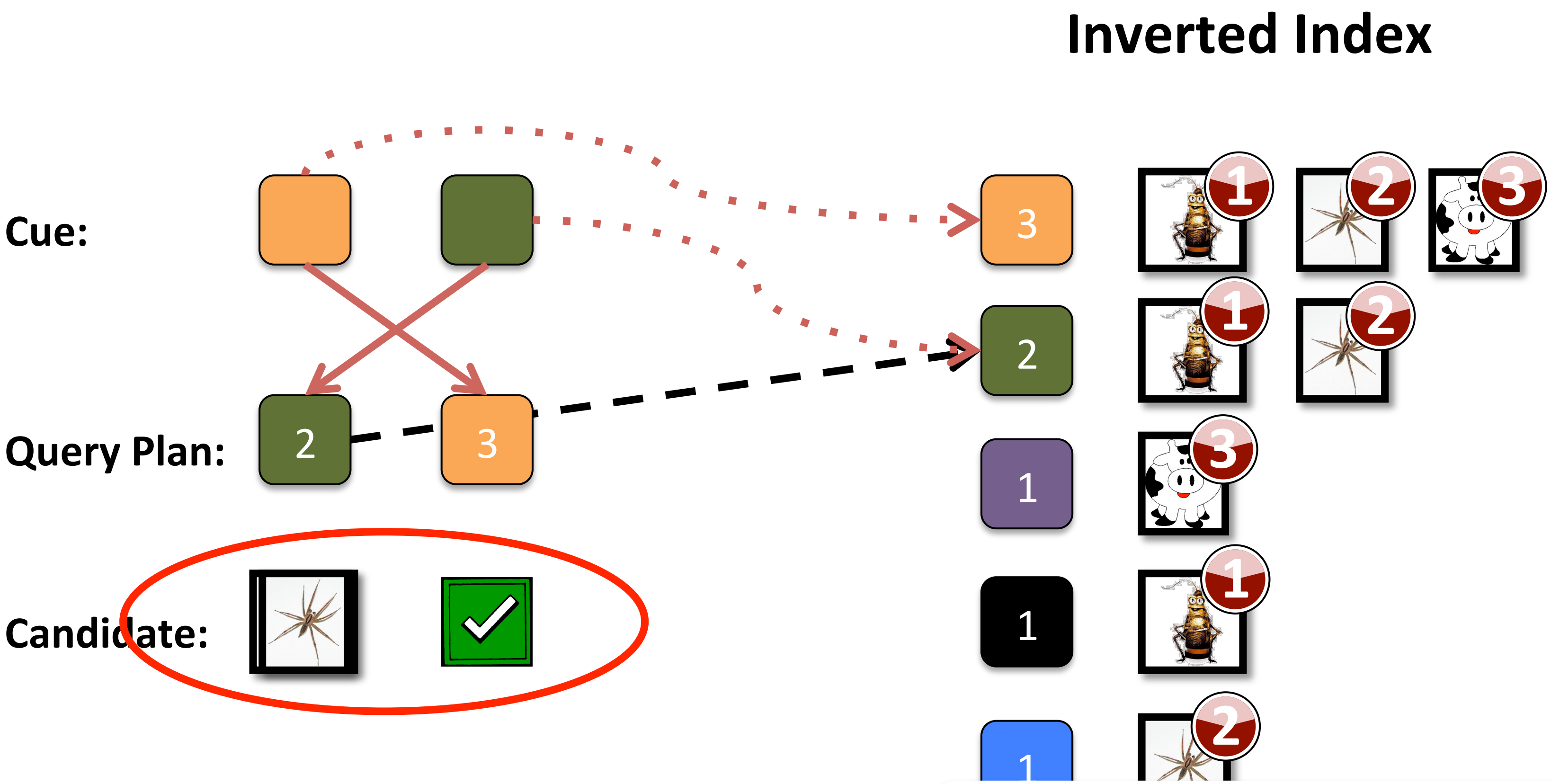
Biased Retrieval Algorithm #2

Static Sort



Biased Retrieval Algorithm #2

Static Sort



Each bias-value update scales with feature cardinality!

Our Hybrid Approach

Empirically supported cardinality threshold, θ

If (cardinality > θ): Sort on Query [#1]

- Candidate enumeration scales with # of objects with large cardinality (empirically rare)

If (cardinality ≤ θ): Static Sort [#2]

- Bias updates must be **locally efficient**
  - Objects affected: $O(1)$
  - Computation: $O(1)$

Empirical Evaluation

Performance Characterization

- Selectivity + Co-occurrence

  O(Failed Candidates)

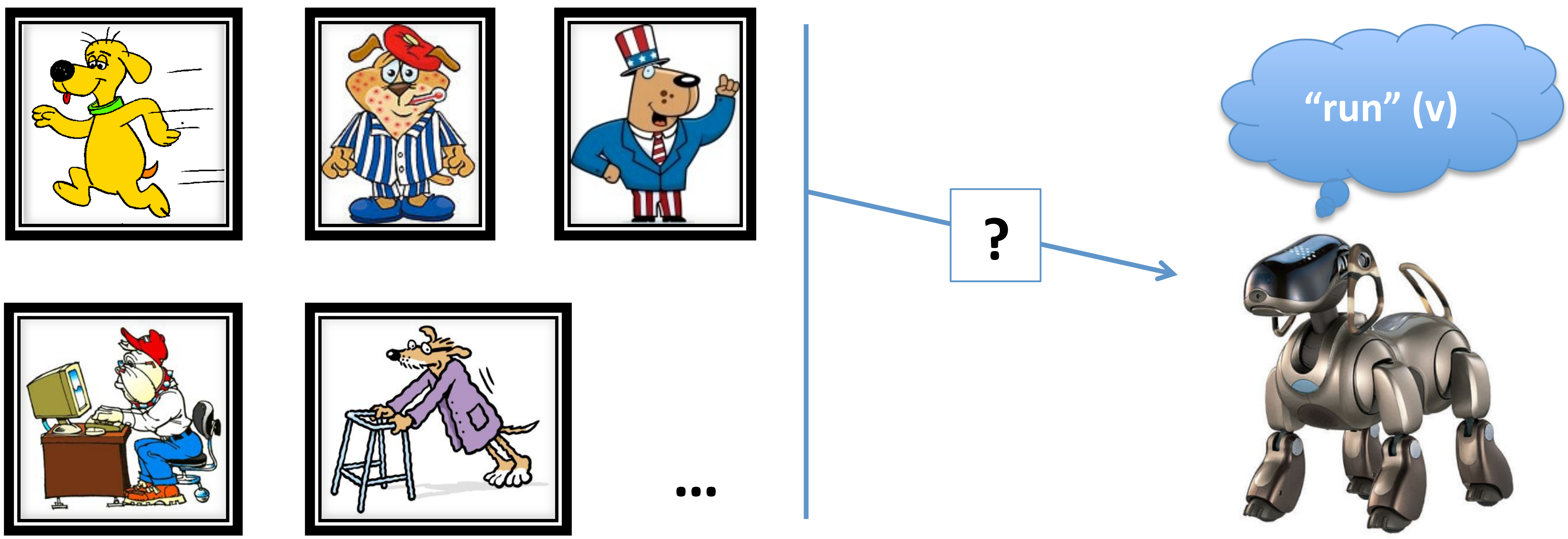
Tasks

- Synthetic: efficiency/scaling of cue matching 
- WSD: efficiency/usefulness of biased retrievals

WSD Evaluation

Motivation

Memory **Agent**



Problem. Ambiguous Cues
Hypothesis. Retrieval History is Useful
Application. Word Sense Disambiguation

WSD Evaluation

Historical Memory Retrieval Bias

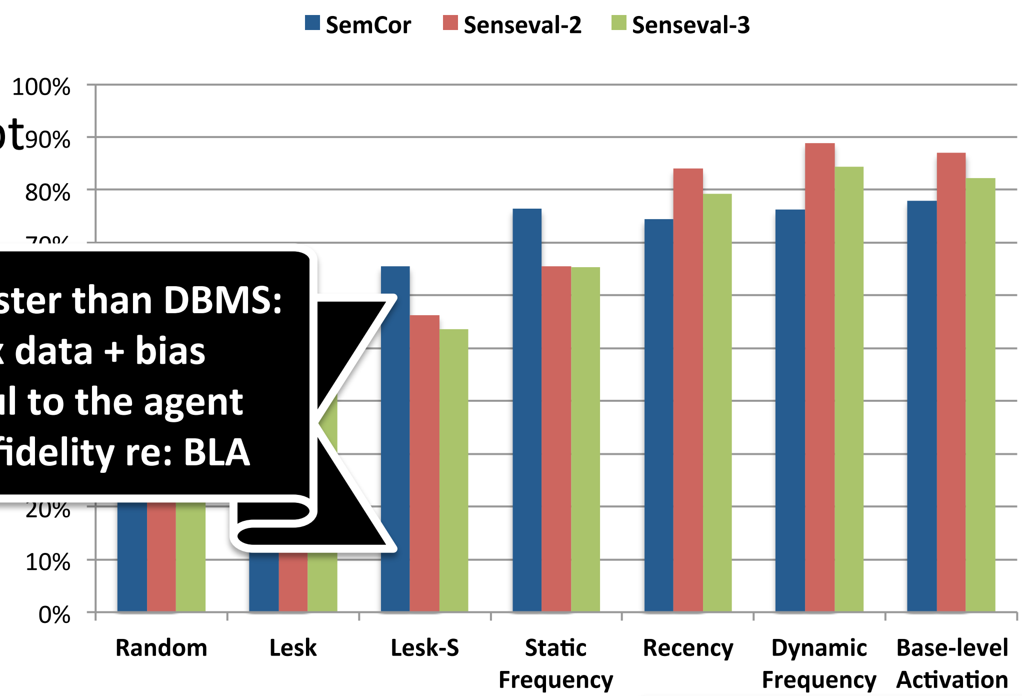
Experimental Setup

- Input: “word”, POS
- Given: WordNet v3
- Correct sense(s) after each attempt

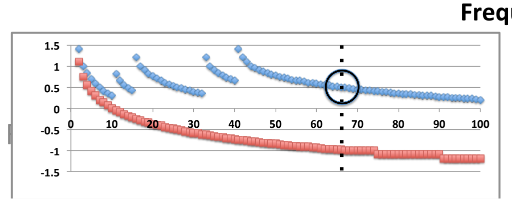
Efficiency & Scaling

- R/DF: $O(1)$, ≤ 0.87 msec.
- Base-Level Activation:
  - Naïve: O(# cand’s), ≤ 13.
  - *Locally Efficient* Appro :
    $O(1)$, ≤ 1.34 msec.
    Idea: relative ranking is all that is required, so re-compute {k^{th}} recent memory as a heuristic (>90% fidelity)

Task Performance (**2nd corpus exp.**)

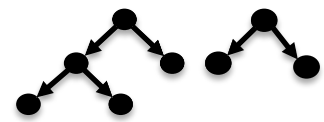


>30x faster than DBMS:
>3x data + bias
Useful to the agent
High-fidelity re: BLA



Biased Retrievals

Semantic Memory

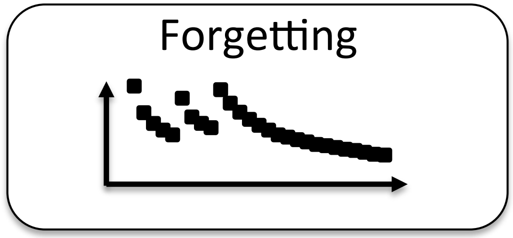


- Algorithms that are reactive and scalable for real tasks and KBs
- Performance characterization w.r.t. general properties of environments, tasks, and agents
- Bias functions that are efficient, scalable, and useful for heuristic reasoning

Ongoing Research

- Prospective memory (Li et al. '12)
- Incremental language processing (Lonsdale et al. '12)
- Mixed-initiative situated instruction (Mohan et al. '13)
- Incorporating likelihood, context
- Consolidation/automatic storage

...



Problem. Extended tasks that involve learning large amounts of knowledge can lead to performance degradation in existing systems (e.g. Kennedy & Trafton, 2007).

Approach. Selectively retain learned knowledge.

Challenge. Balance…

– maintenance of high task performance
– reduction of computational resources across a variety of tasks.

Hypothesis

Rational to forget a memory if…

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

Evaluation. 2 complex tasks, 2 memories

Mobile Robot Navigation

Working Memory

- bounds decision time
- completes task
  - ➢ 1 hour

Multi-Player Dice

Procedural Memory

- 50% memory reduction
- competitive play
  - ➢ days

Forgetting: Naïve Approach

Algorithm

- At each time step
  - For each memory element
    - If (Activation < Threshold)
      - » Forget

Efficiency Evaluation

- Per Time Step: O(|**Memory Elements**|)

Efficient Forgetting via Decay Prediction

<u>Algorithm</u>

- On new activation event
  - *Predict* time of future decay
  - Add to *time-keyed map*

- At each time step *t*
  - Remove elements in map at key *t*

<u>Complexity Analysis</u>

Per Time Step: O(|Decayed| + |Events|*[Prediction Cost])

Decay Prediction

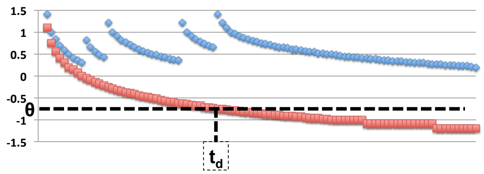
Efficient and Correct

1. Cheaply approximate decay on each access
   – Underestimate time of decay by treating each time step of memory access independently: $O(1)$

2. Exact determination
   – Binary parameter search: $O(\log_2 T)$
   – <u>Not needed</u> if element is removed by #1 estimate
   – Otherwise, <u>reduced</u> by the degree to which #1 is accurate

Novel Base-level Decay Approximation

Given



constants

- Decay threshold (θ)
- Decay parameter value (*d*)

and a set of *n* memory accesses…

- Time steps since access (*s*)
- Number of accesses (*k*) at that time step

solve for…

- Time steps (t_d) till memory decay

Calculation

For each memory access…

$$\ln(k \cdot [t+s]^{-d}) = \theta$$

$$\ln(k) - d \cdot \ln(t+s) = \theta$$

$$\ln(t+s) = \frac{\theta - \ln(k)}{-d}$$

$$\boxed{t = e^{\frac{\theta - \ln(k)}{-d}} - s}$$

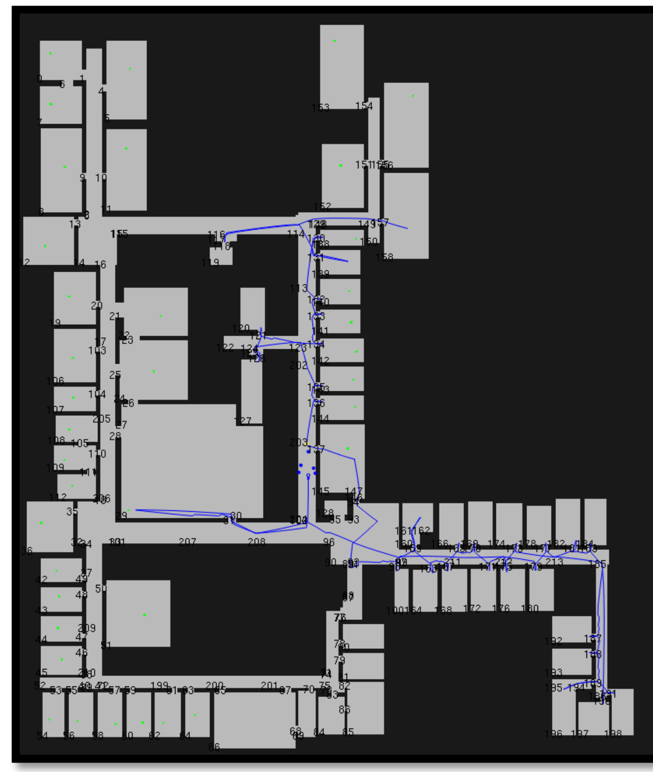
$$t_d >= \sum_{j=1}^{n} t$$

Task: Mobile Robot Navigation

Simulated Exploration & Patrol

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

– Builds map in memory from experience



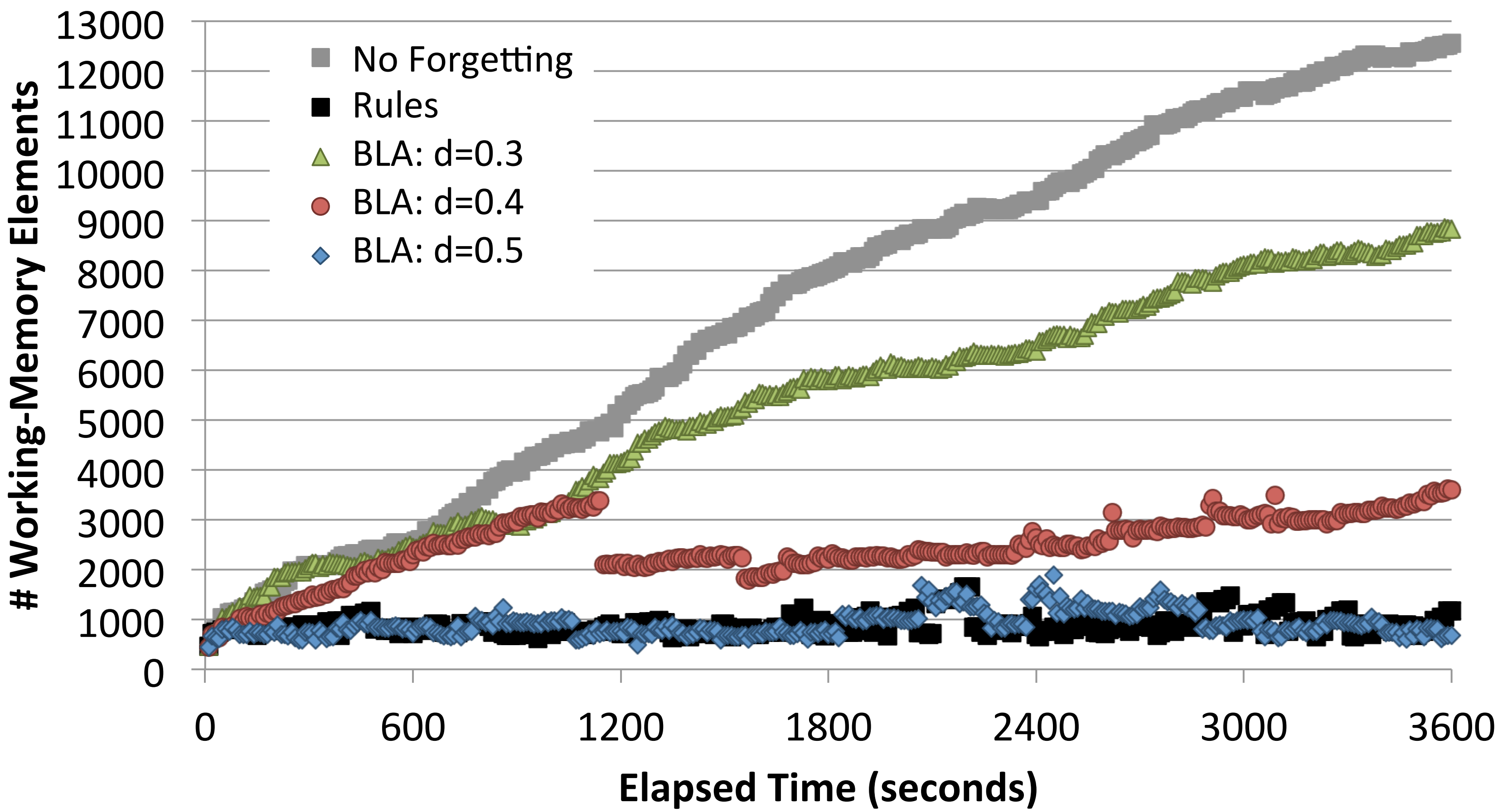
Problem: Reactivity

Issue. Increasing map knowledge in working memory (most used infrequently) ->
large episodes -> long reconstruction time.

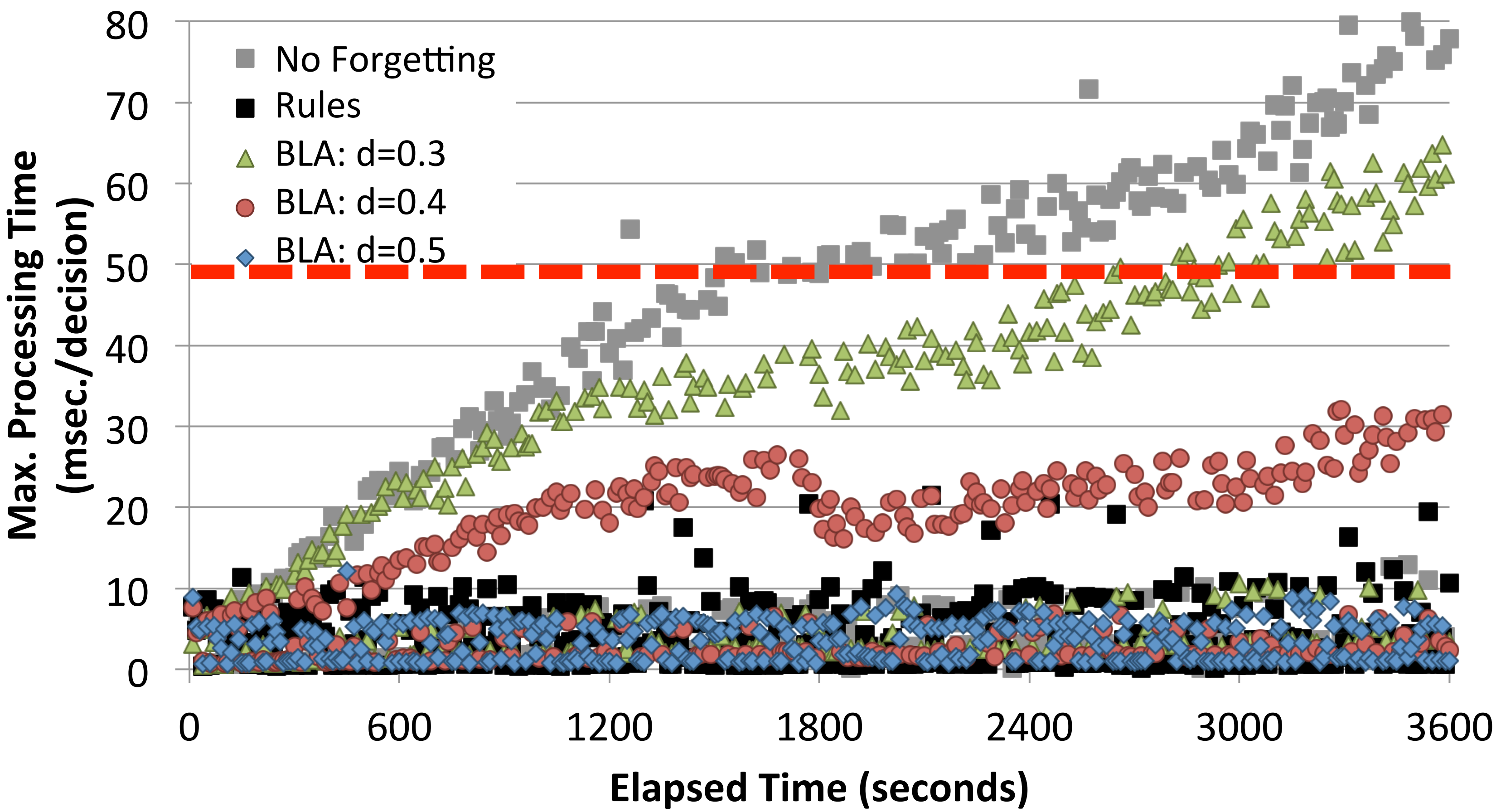
Approach. Task-independent memory hierarchy

1. Automatically forget unused short-term features of long-term objects
2. General knowledge to retrieve from SMem as necessary

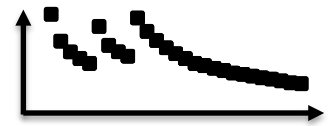
Results: Working-Memory Size



Results: Decision Time



Forgetting

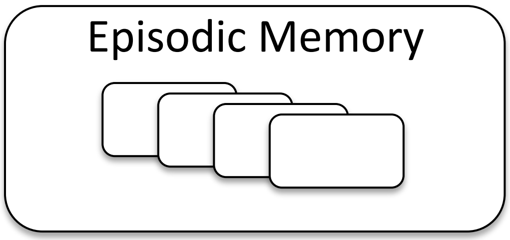


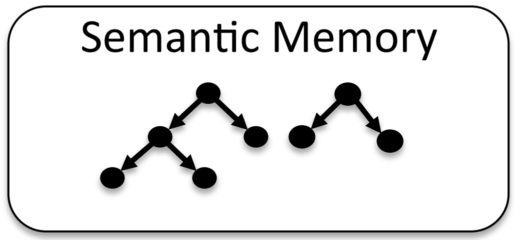
- Explored common forgetting hypothesis in two memories, two complex tasks
- Developed efficient and correct method of forgetting via base-level activation model
- Improves reactivity and scaling for long lifetimes and large amounts of knowledge, without reducing task performance

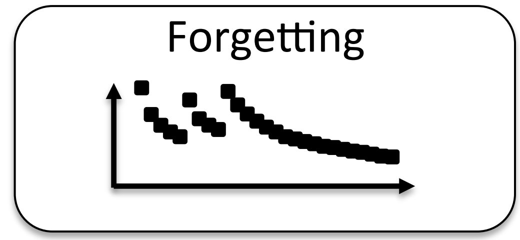
Ongoing Research

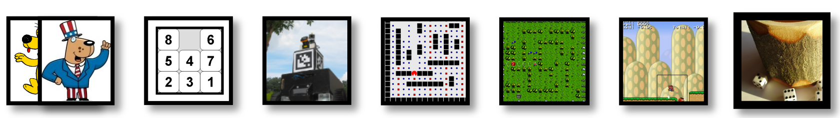
- Bounding storage for long-lived agents & mobile platforms
- Consolidation

Summary







- **Analysis**. Properties of Environment, Task, Agent
  - Algorithms: Efficient, Scalable, Task-Independent
- **Integration**. Soar v9.3.2
- **Evaluation**. 
  - Demonstration of Agent Benefits

Thank You :)

Questions?

John Laird
Professor
Michigan

Georg Essl
Asst. Prof.
Michigan

Justin Li
PhD Cand.
Michigan