Effective and Efficient Memory for Generally Intelligent Agents

Nate Derbinsky

Chair: John E. Laird

Oral Defense

My 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 current 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)

Two Common Mechanisms

Episodic

- Autobiographical
- Embedded in personal context

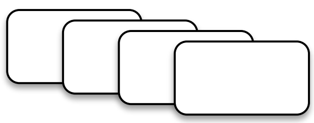
Semantic

- General facts
- Independent of learning context

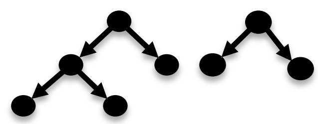
This Dissertation

Effective and Efficient Memory

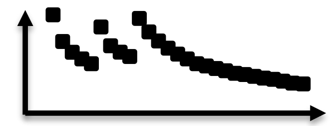
Episodic Memory



Semantic Memory



Forgetting



Desiderata

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

Research Approach

Analysis. Properties of Environment, Task, and Agent
- Identify regularities
- Exploit in algorithms

Integration. Cognitive Architecture
- Apply task-independent knowledge representations
- Implement task-independent processes

Evaluation. Variety of Domains, Large Scale
- Characterize performance via task properties
- Demonstrate benefits to agents

Outline

Architectural Integration

Episodic Memory

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

Semantic Memory

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

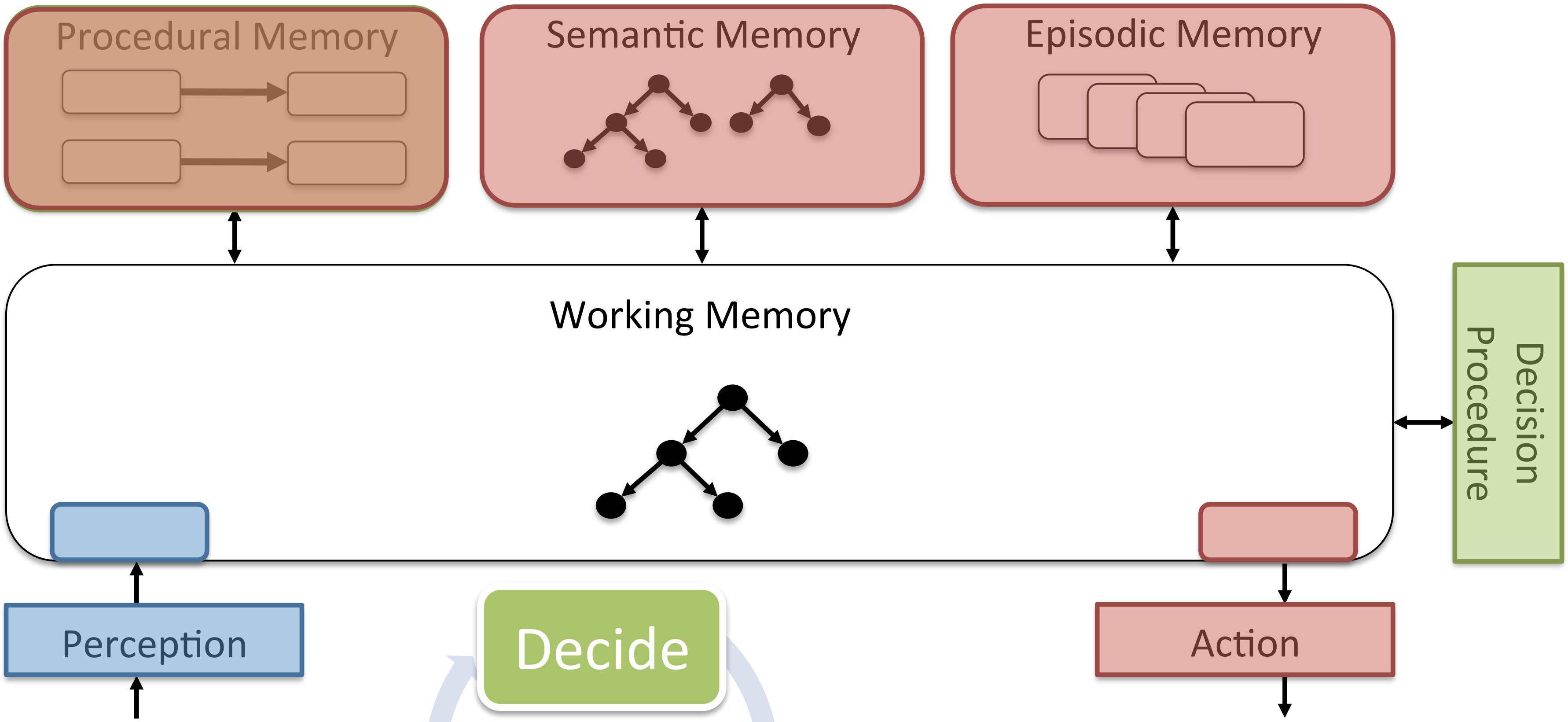
Forgetting

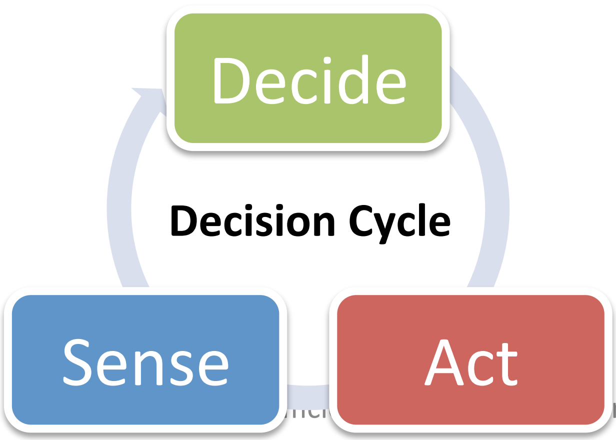
[ICCM ’12a], [ICCM ’12b]

Summary & Discussion

Soar (Laird, 2012)

Memory Integration

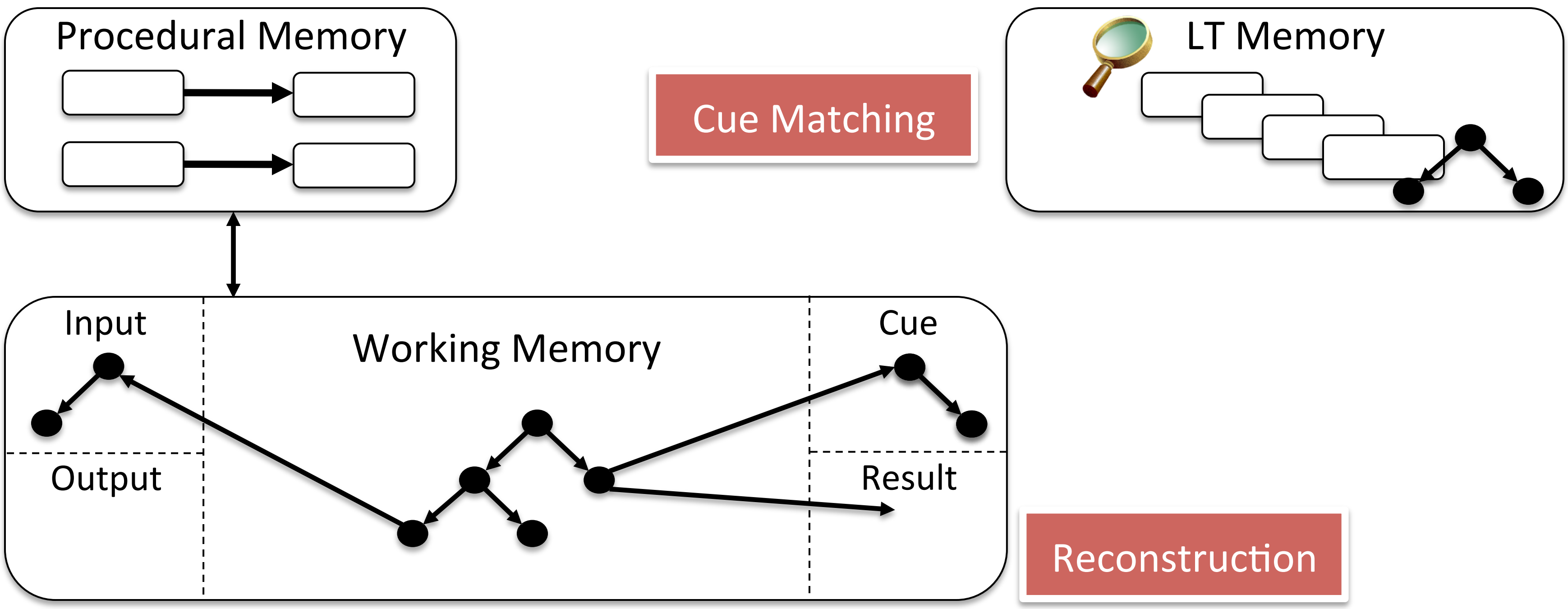




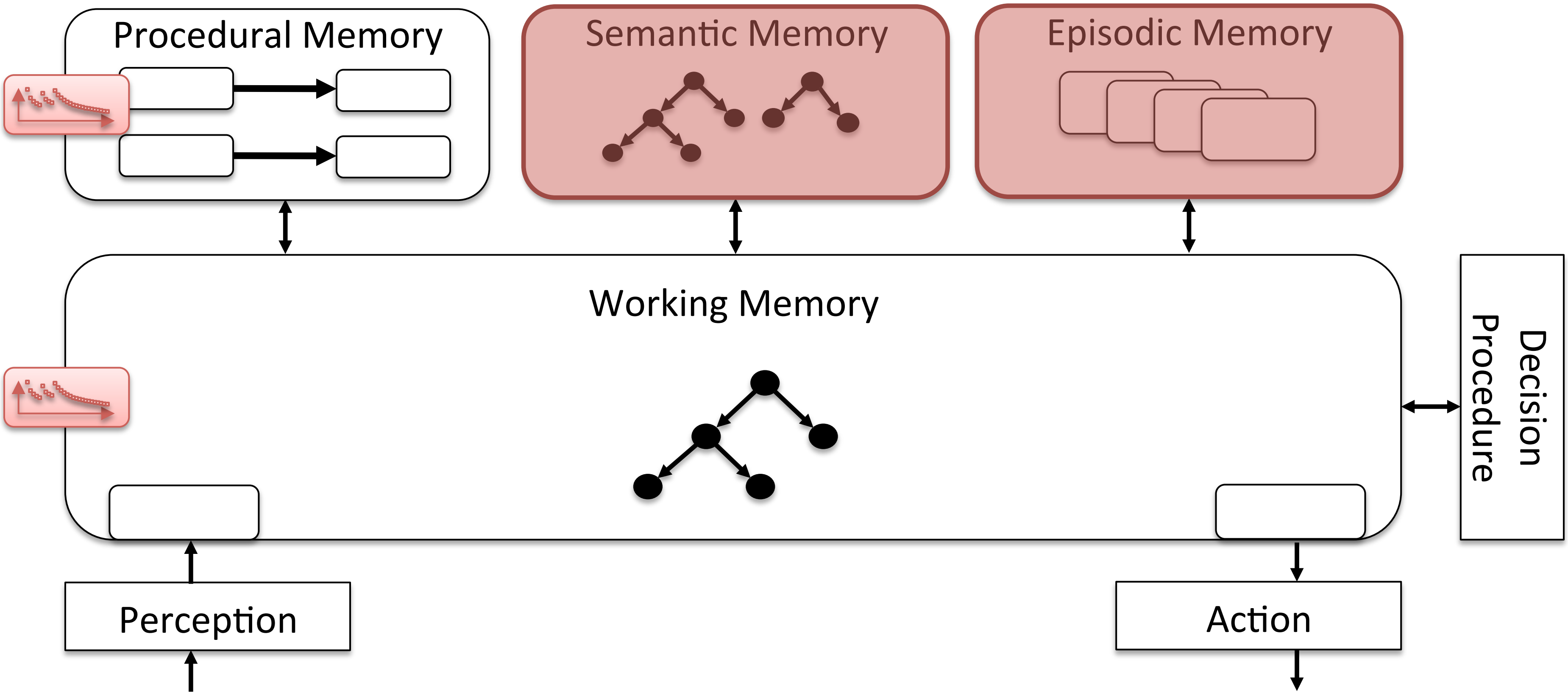
Agent **reactivity** is the time required to execute each decision cycle, <u>including</u> LT memory access ∴ only bounded search!

Soar

LT Memory Access



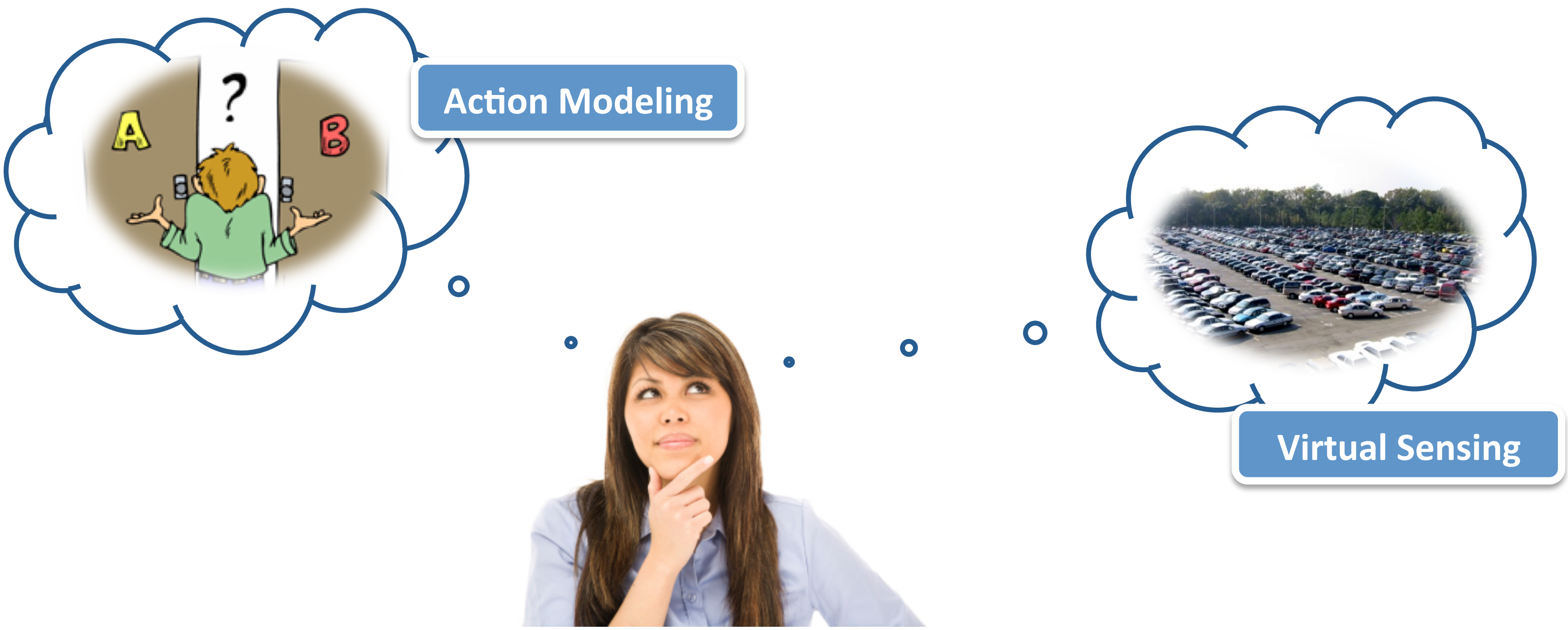
This Dissertation



Open Source in Soar v9.3.2

Episodic Memory

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

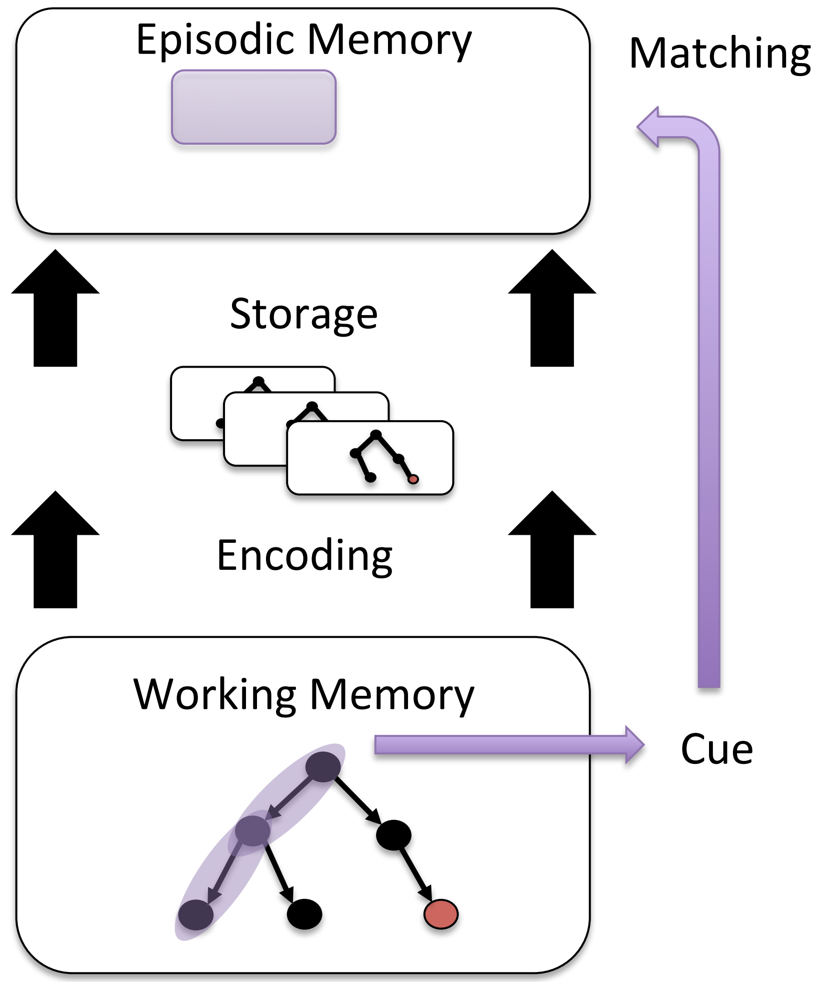


Episodic Memory
Problem Formulation

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

Encoding/Storage
- Automatic
- No dynamics

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



Episodic Memory

Computational Challenges

Arbitrary, dynamic state

Scaling potential, agent…

- state (1000s nodes/edges)
- life (10^6-10^9 episodes ~ days)

Cue-matching optimality

- Constrained subgraph isomorphism (NP-complete)
- Search: *O*(# episodes)

Analysis & Algorithms
[ICCBR '09], [AAMAS '12]

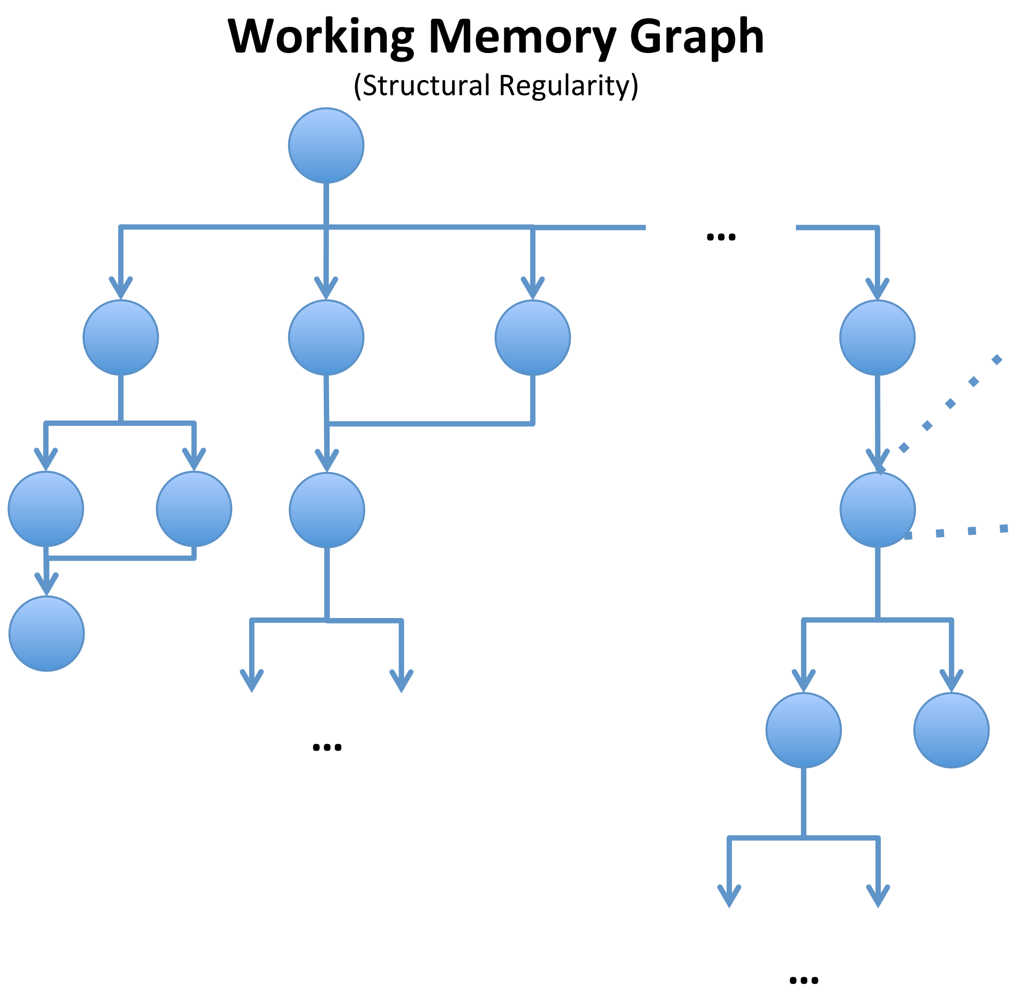
Properties

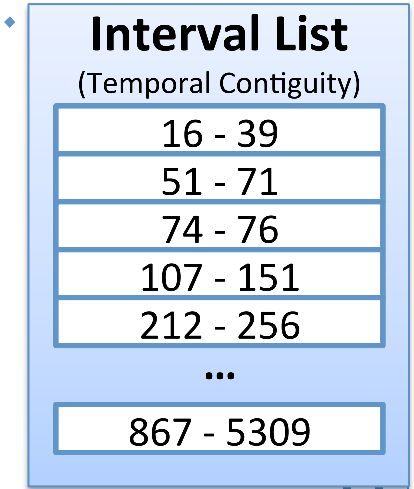
- Temporal Contiguity
  |state changes| << |state|
- Structural Regularity
  |distinct structures| << |all experienced structures|

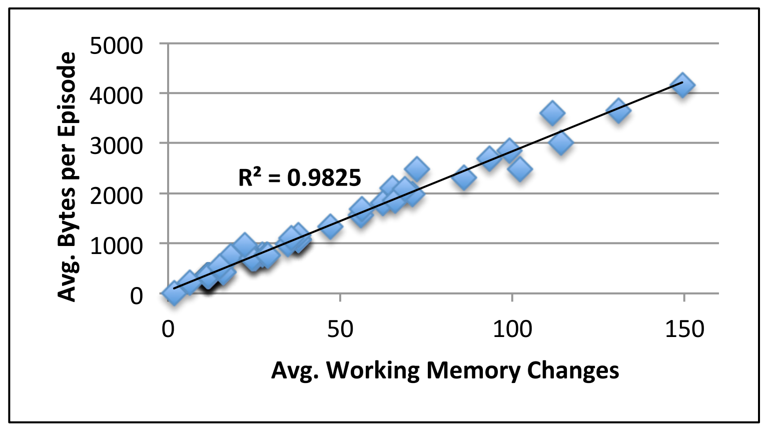
Algorithms

- **Storage**: dynamic graph index*
- **Cue Matching**: 2-phase search*
- **Reconstruction**: relational interval tree
  (Kriegel et al. 2000)

Dynamic Graph Index

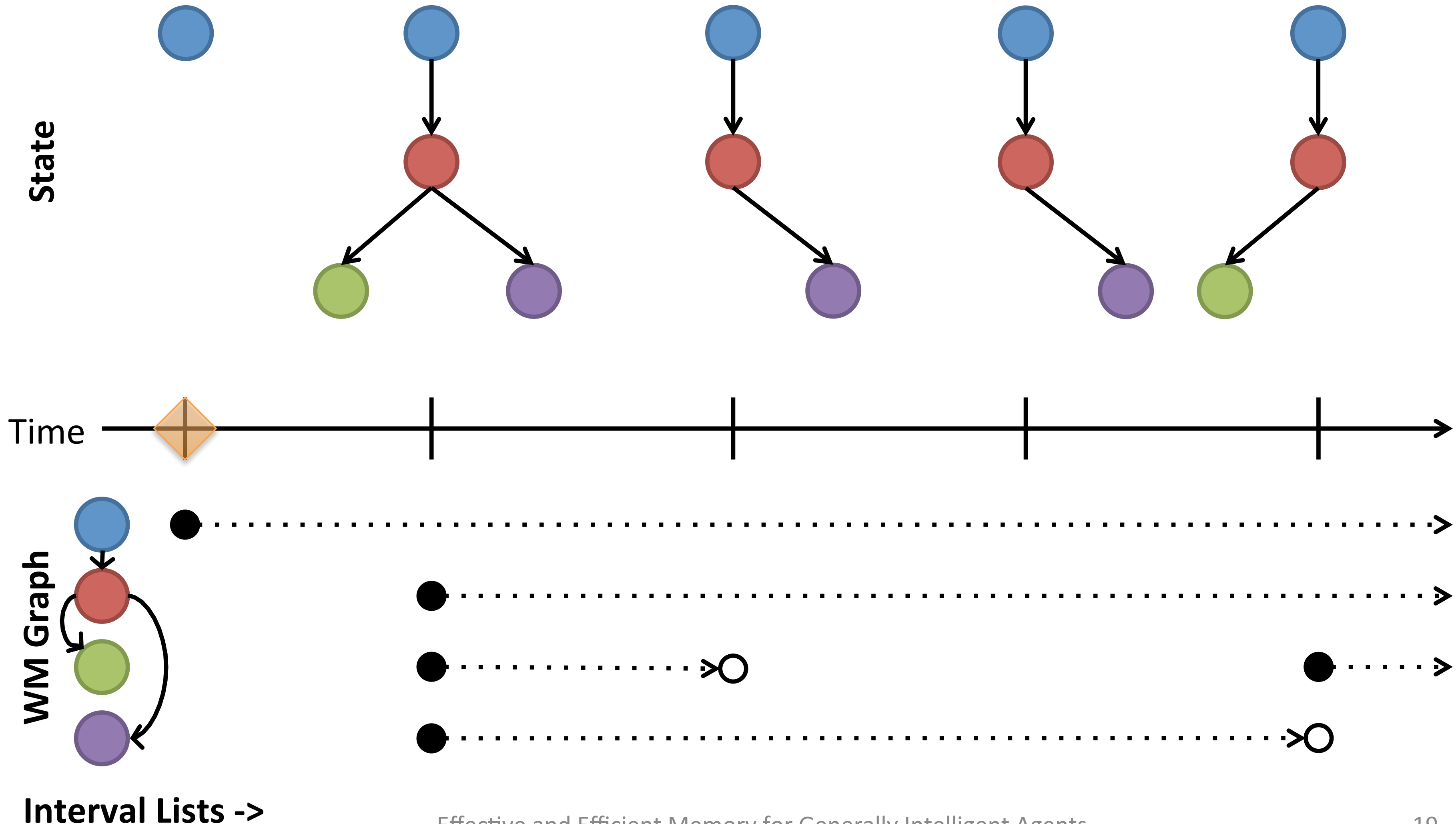






Incremental Encoding Algorithm

Only Store Changes



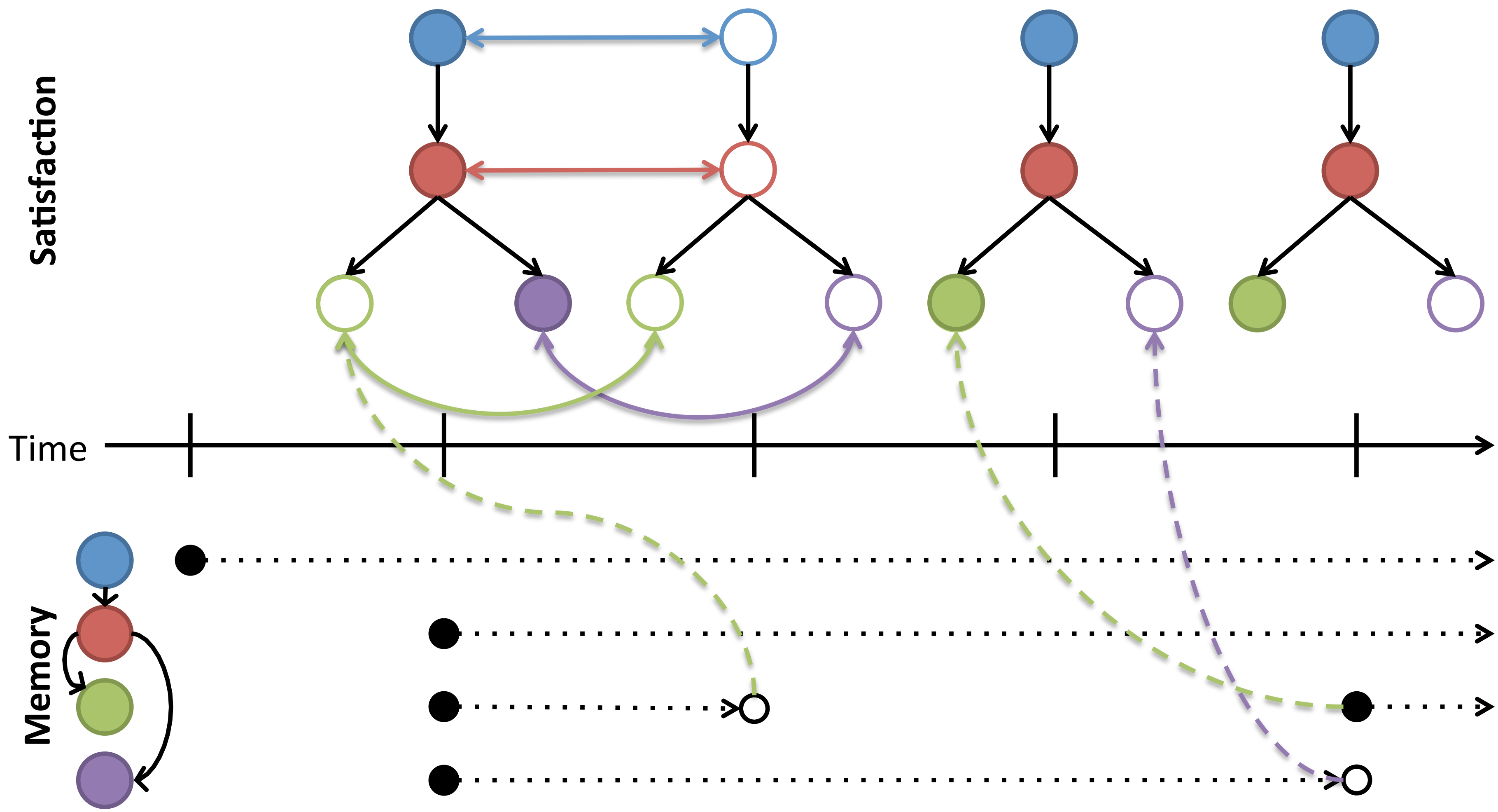
2-Phase Cue Matching

1. **Surface**
   a) Identify cue-feature changes via ordered interval-walking algorithm
      - Priority queue of b+-tree pointers

   b) Incrementally score features independently
      - Discrimination network (DNF Graph)

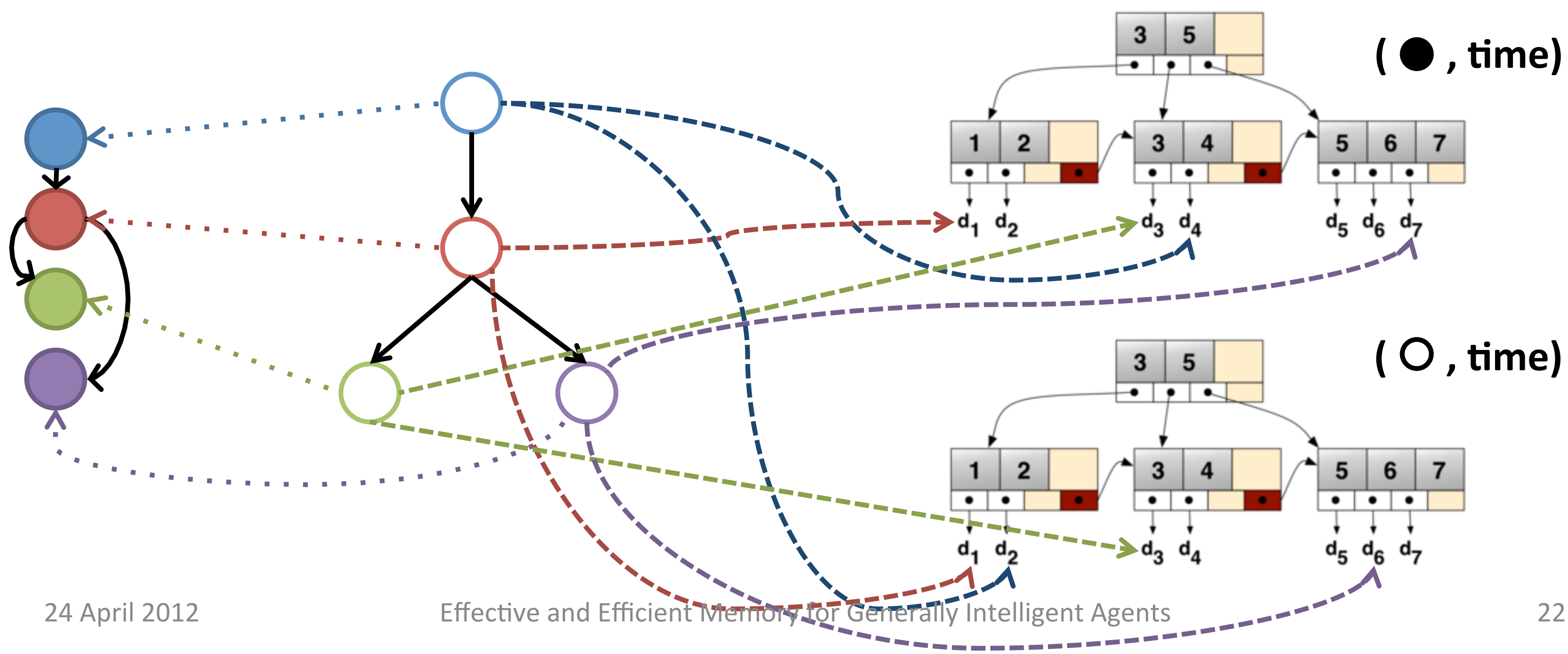
2. **Structure**
   a) Graph match + standard heuristics (e.g. MCV)

Cue Matching Algorithm



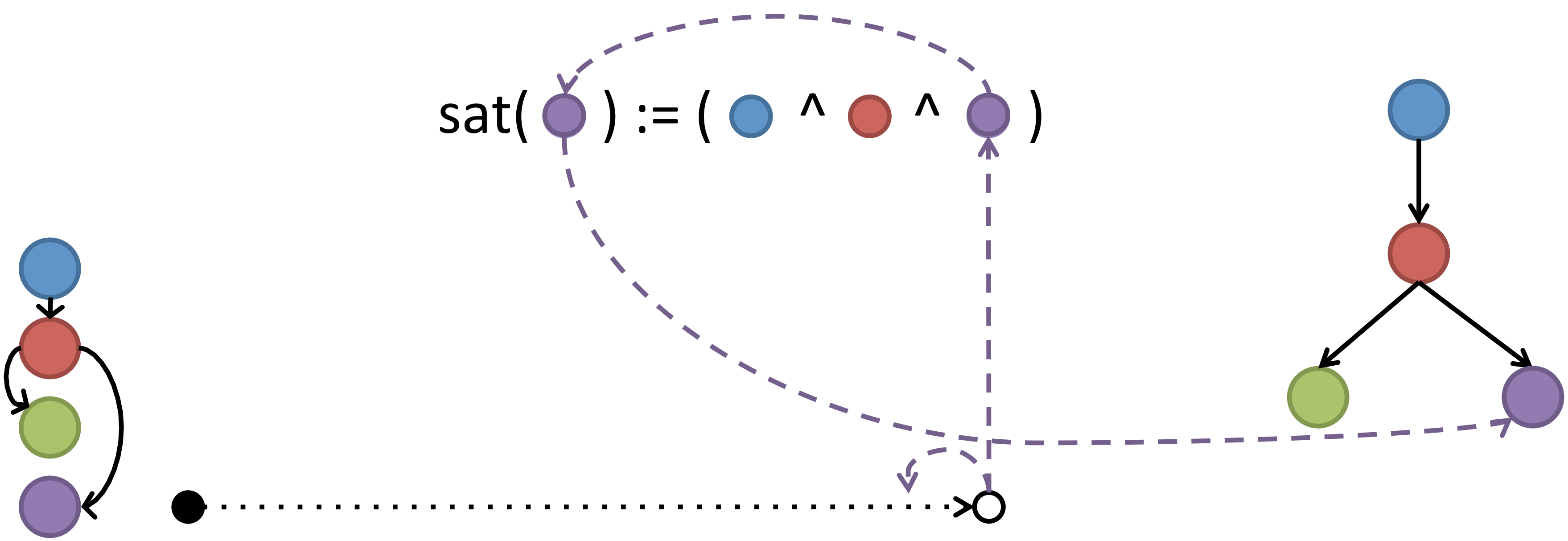
Details: Interval Walk

- Maintain interval endpoint sorting via b+ trees
- On cue, add leaf pointers to time-keyed p-queue
  - Pop as necessary to process ● or ○



Details: Incremental Episode Scoring

- Cue edges serve as minimal propagation directives
  - Maps to DNF SAT: sat(n) := sat(n) ^ sat(par(n))
- On ●/○, update literal(s), clause(s), possibly recurse

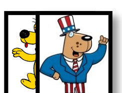
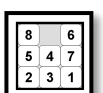


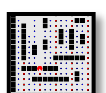
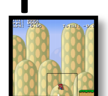
Empirical Evaluation
[AAAI ’12a]

<u>Performance Characterization</u>

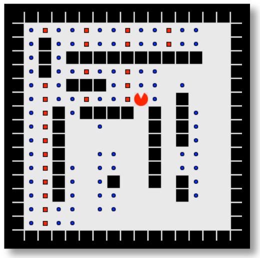
- Temporal Selectivity + Co-Occurrence
  O(Search Distance)
- Structural Selectivity
  O(Episode Hyper-edges)

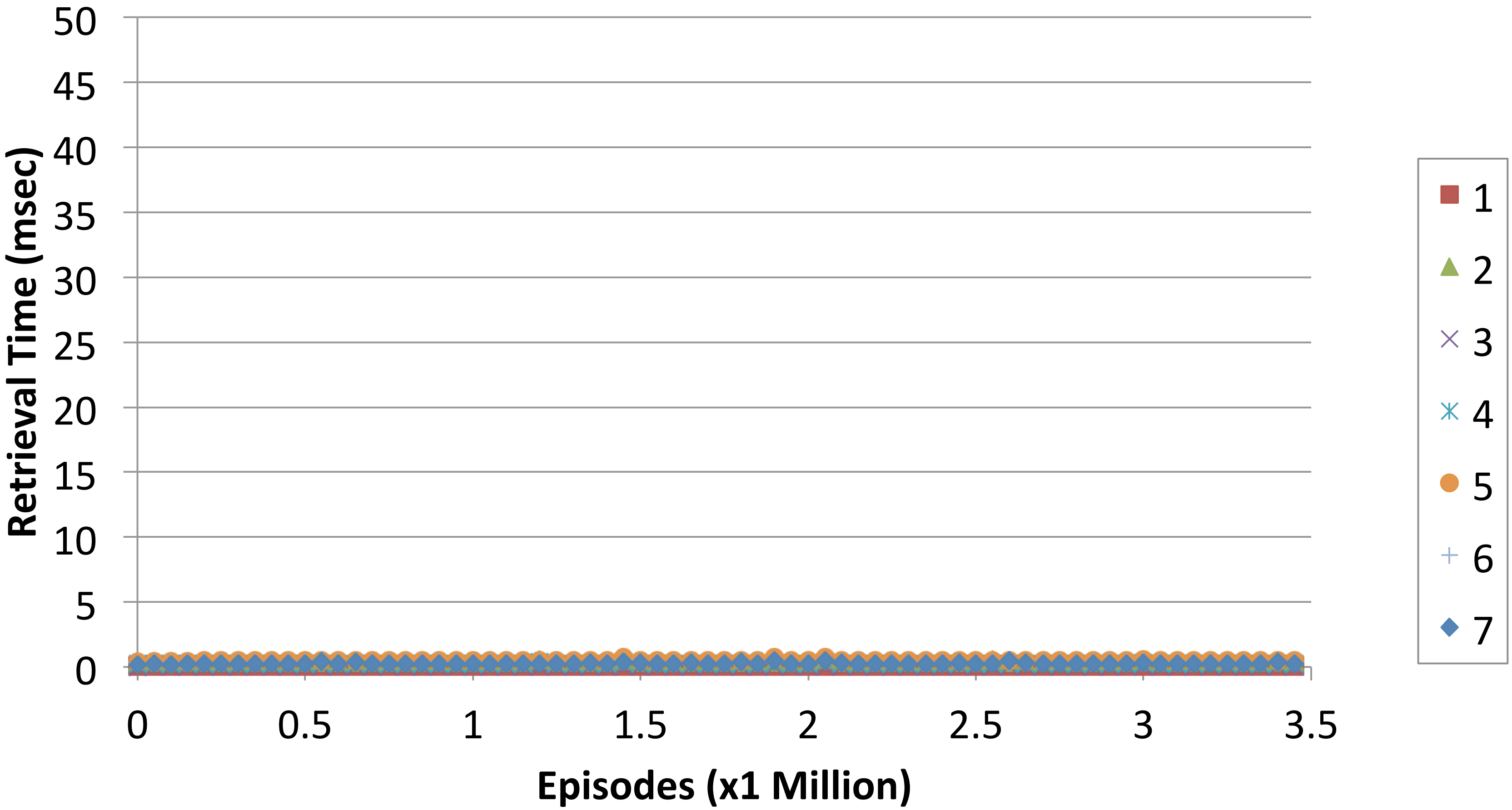
<u>Empirical Evaluation</u>

- 49 domains: WSD, planning, robotics, games
   

     
- 10^5-10^8 episodes ~ days of real time, >100 cues

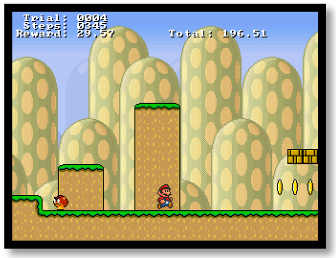
Data: Eaters

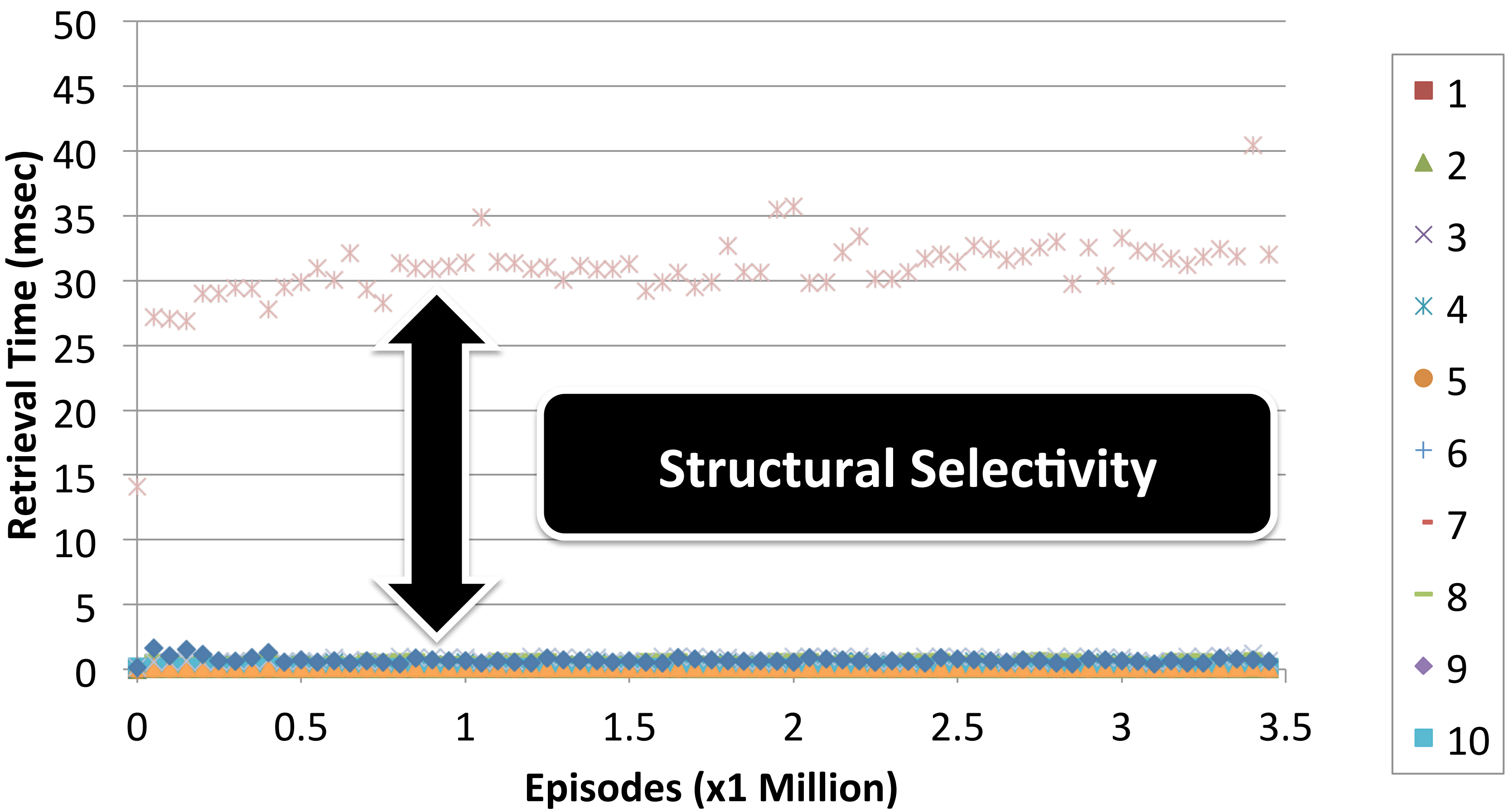




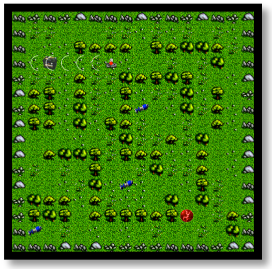
Retrieval Time (msec)
50
45
40
35
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
Episodes (x1 Million)
1
2
3
4
5
6
7

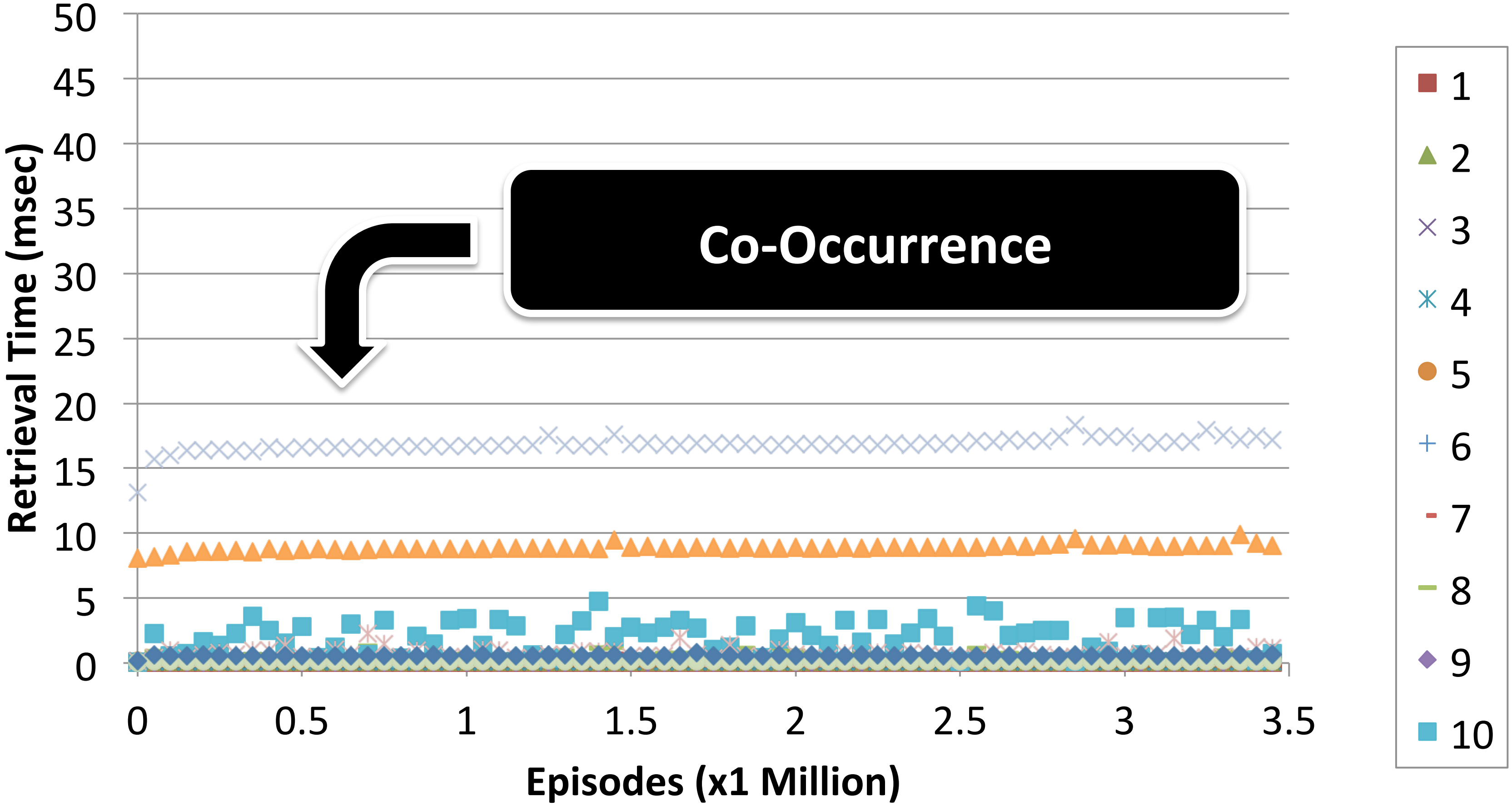
Data: Infinite Mario



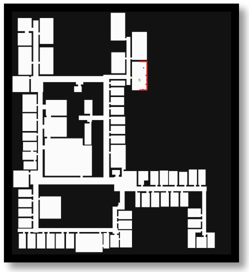


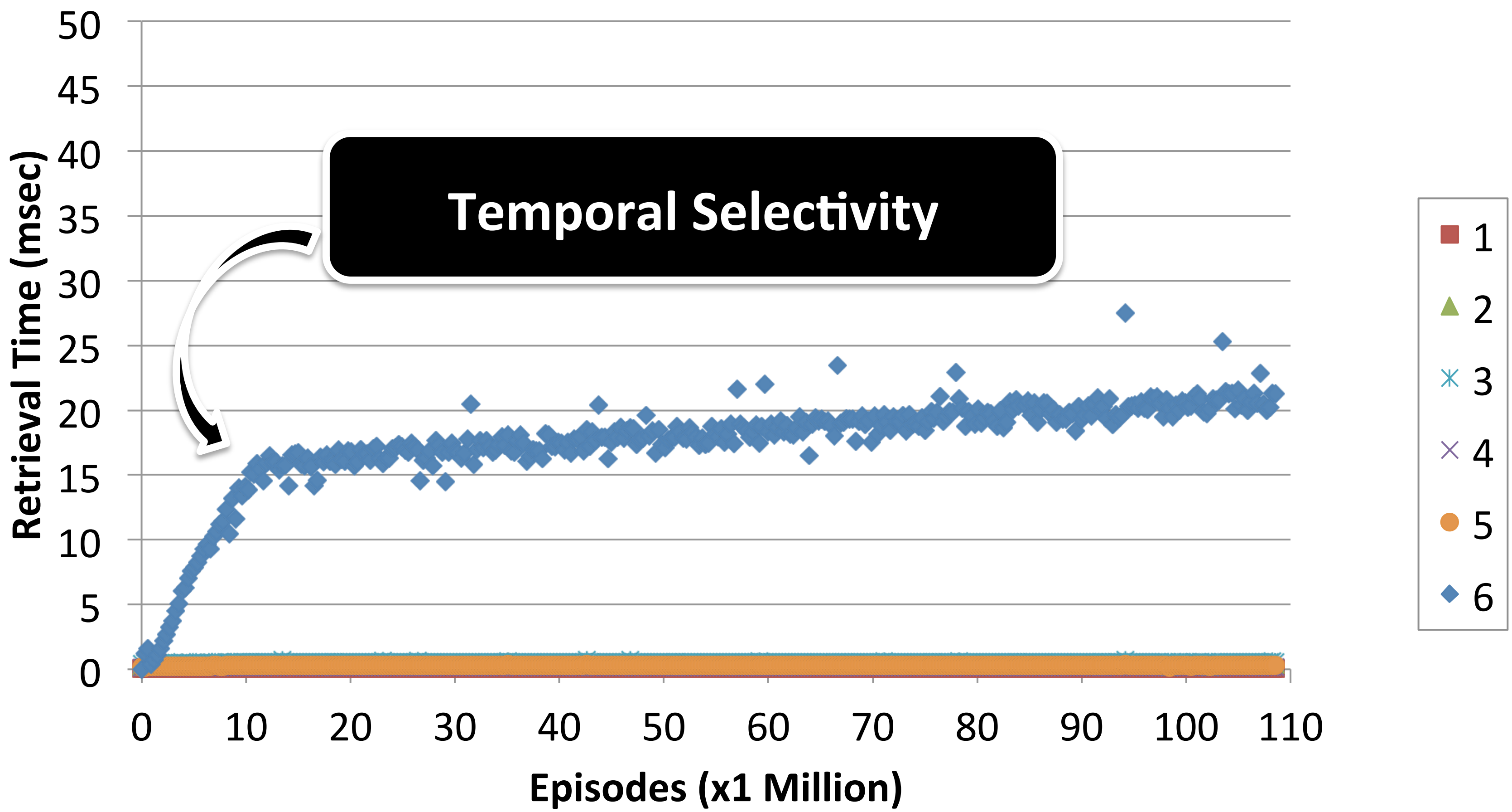
Data: TankSoar





Data: Mobile Robotics



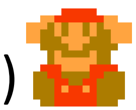


Evaluation Results

Generality

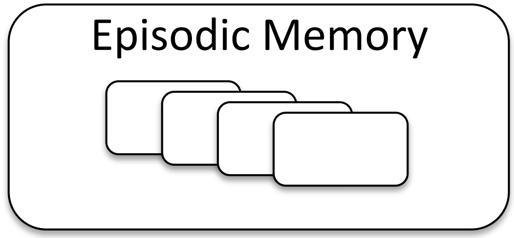
- Demonstrated 7 cognitive capabilities
  - Virtual sensing, action modeling, long-term goal management, …

Reactivity

- <50 msec. storage time for all tasks (ex. temporal discontiguity) 
- <50 msec. cue matching for many cues

Scalability

- No growth in cue matching for many cues (days!)
  - Validated predictive performance models
- 0.18 - 4 kb/episode (days – months)



- 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

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
Problem Formulation

Representation

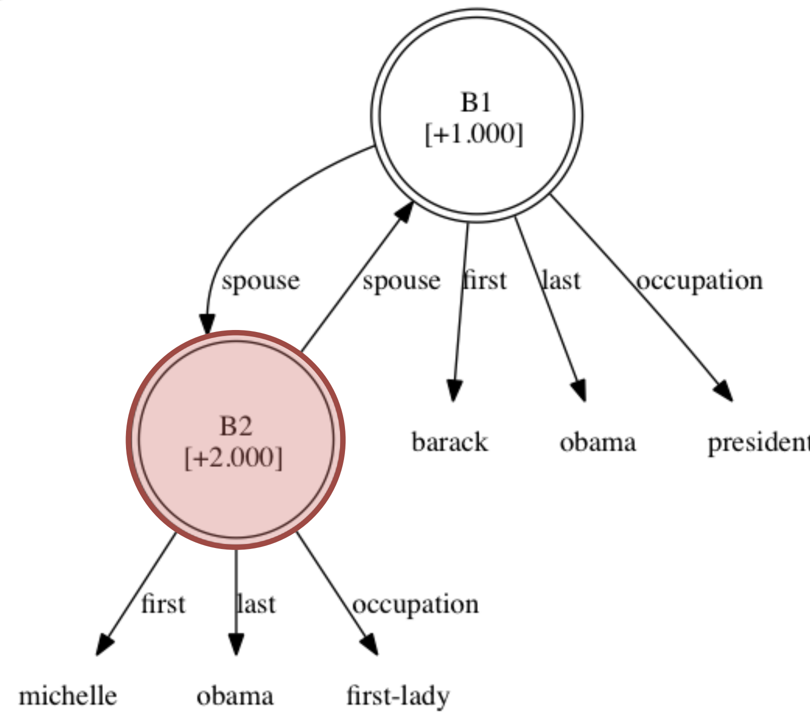
- Directed graph

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…

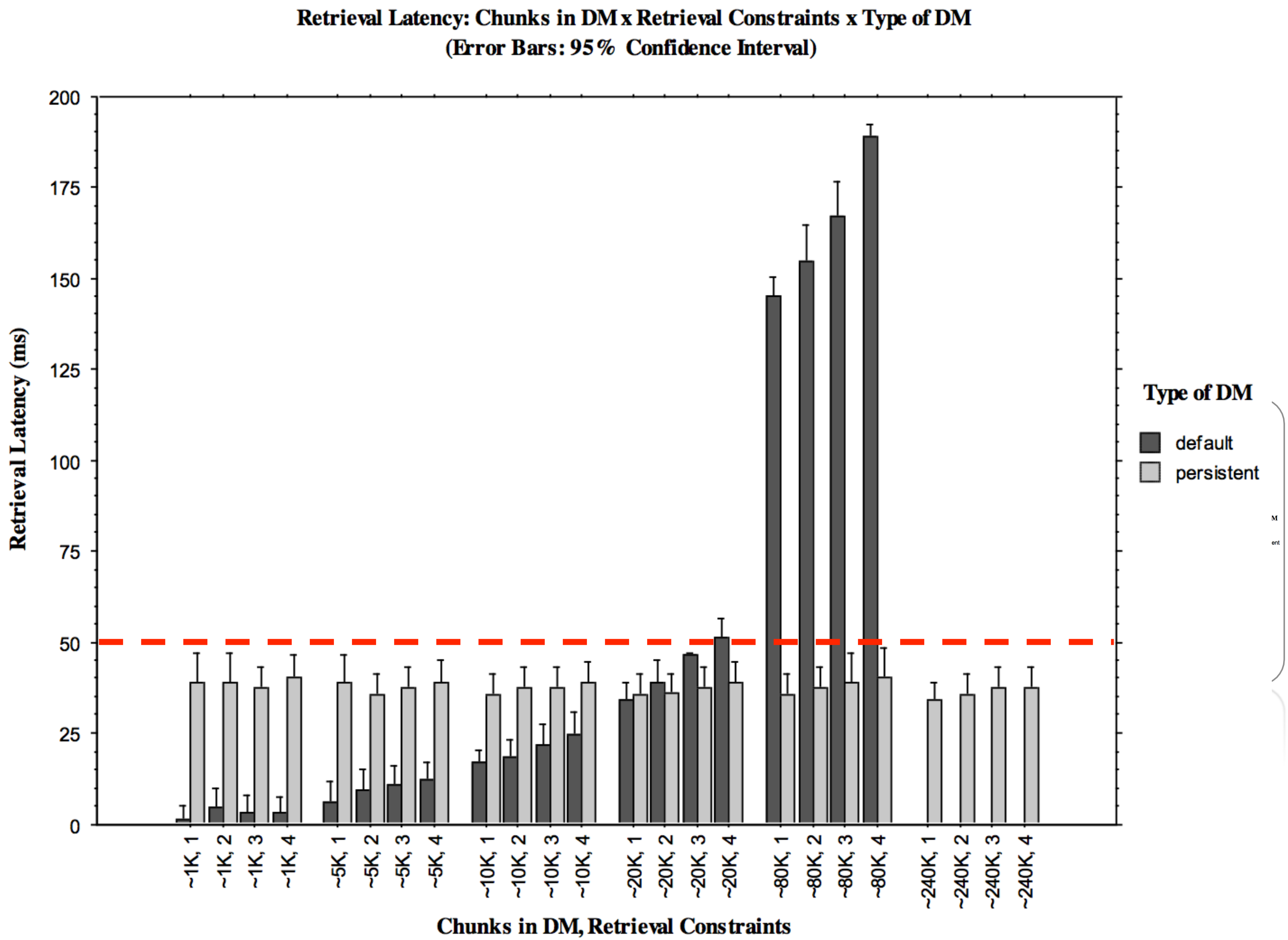
- number of nodes/edges
- symbol vocabulary

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
[ICCM ‘10], [AAAI ‘11]

<u>Properties</u>

- Object Cardinality

  Few objects with large # of features/relations

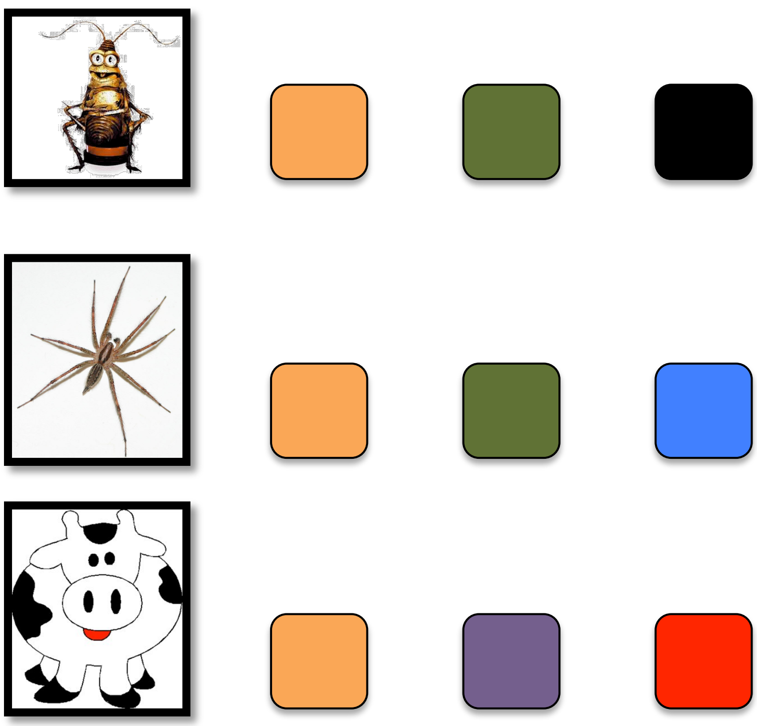
<u>Algorithms</u>

- **Storage**: incremental inverted index (b+ trees)

  (Zobel and Moffat, 2006)
- **Cue Matching**:
  - Statistical query optimization (Chaudhuri, 1998)
  - Hybrid ranking via *locally efficient* bias functions*

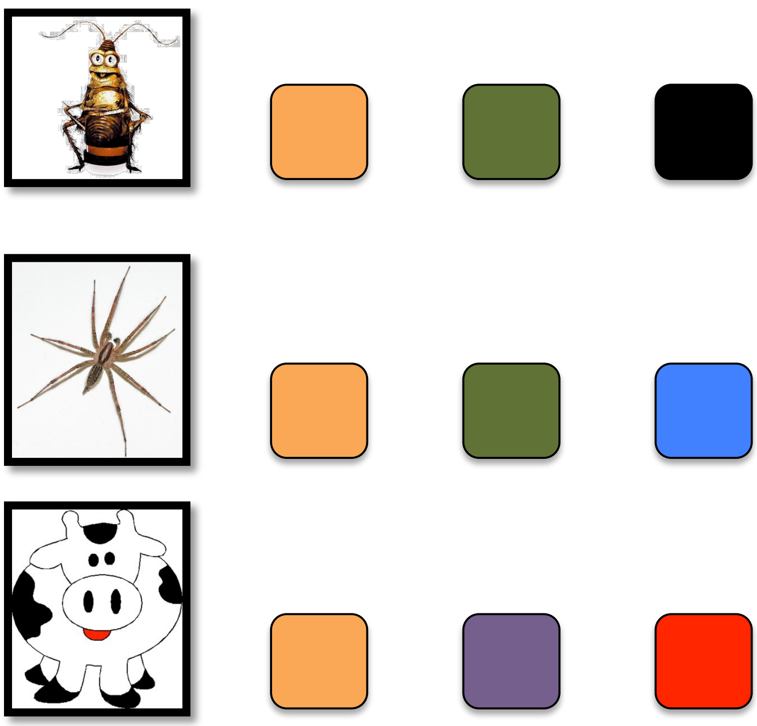
Example Semantic Knowledge

Semantic Objects: Features

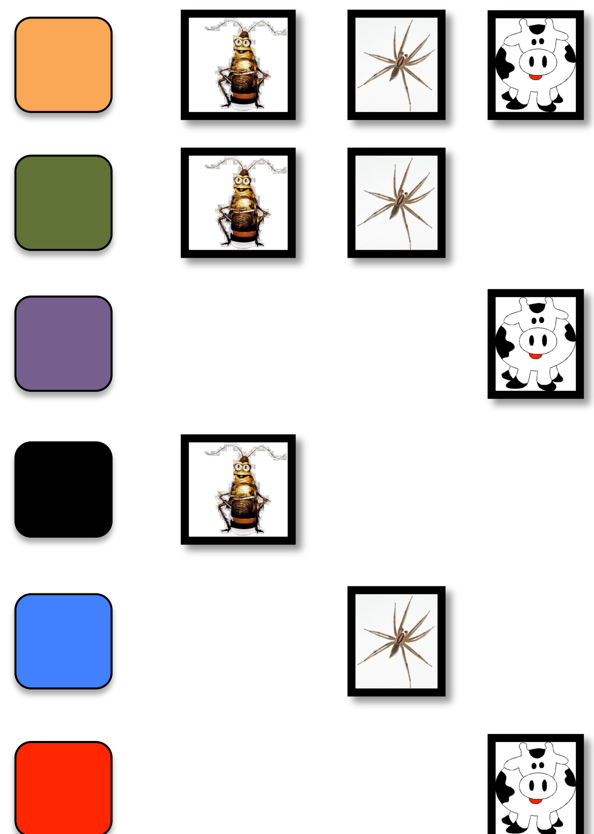


Inverted Indexing

Semantic Objects: Features

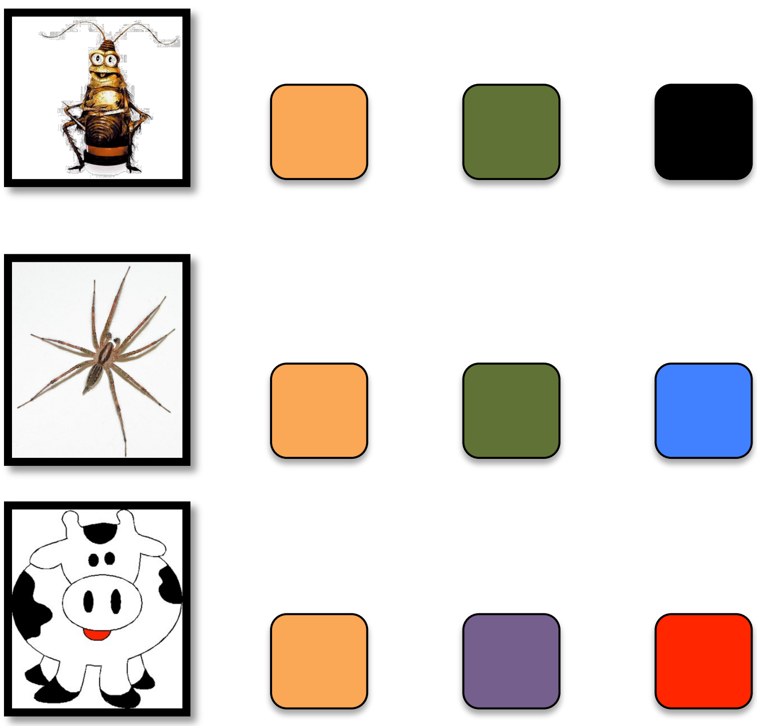


Inverted Index

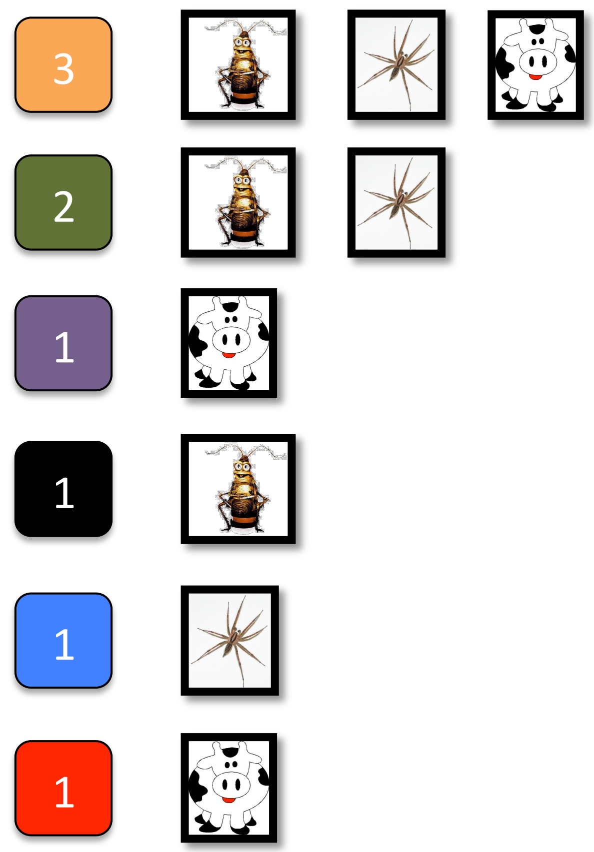


Feature Statistics

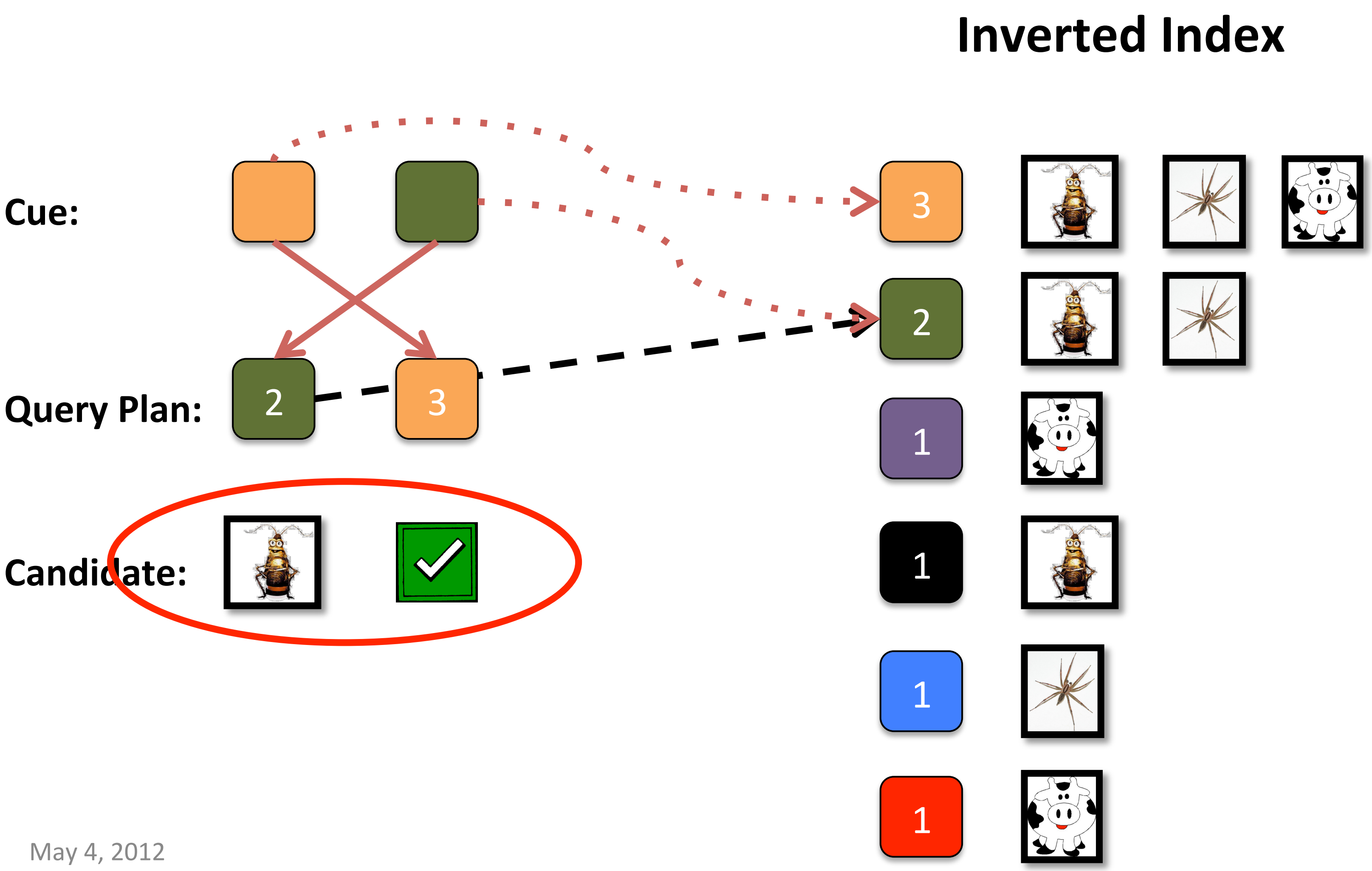
Semantic Objects: Features



Inverted Index

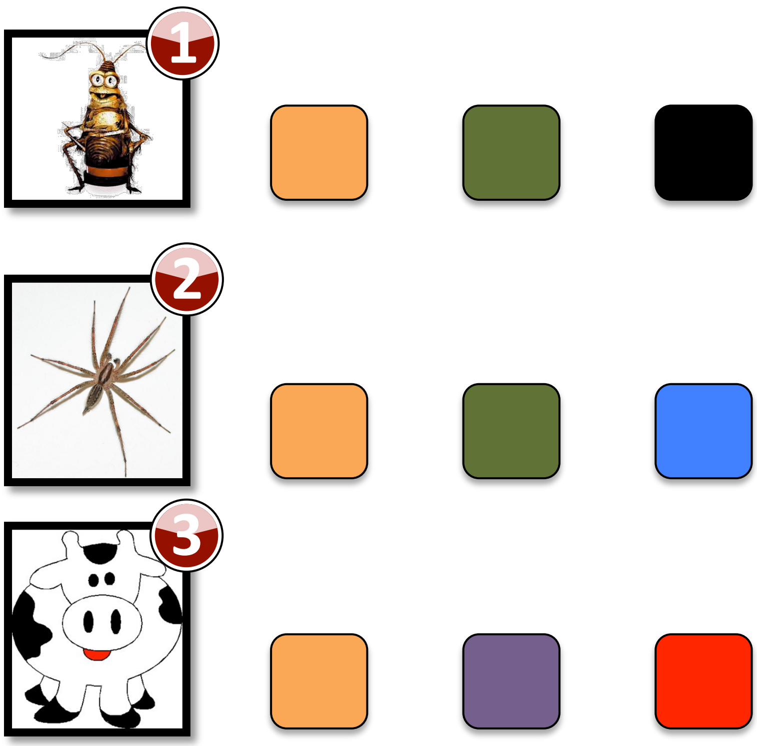


Non-Biased Retrieval Algorithm

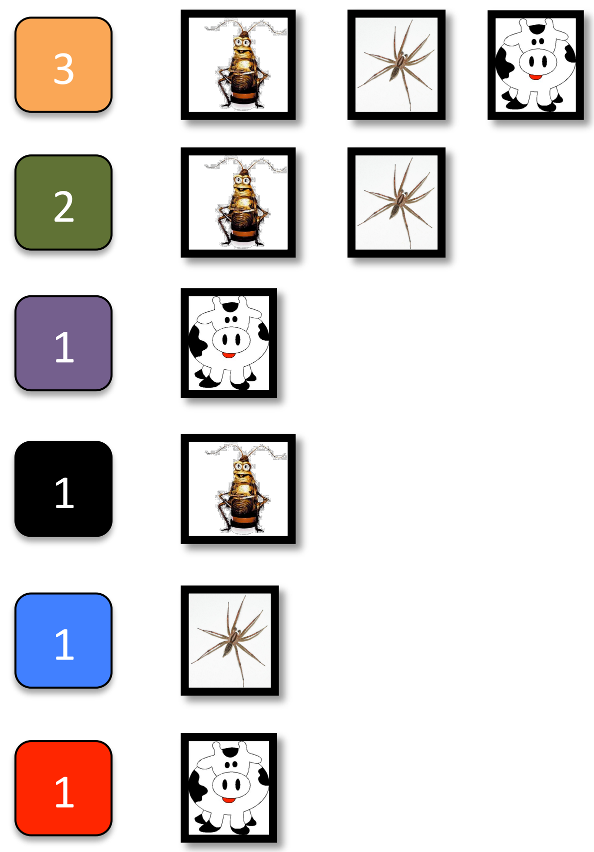


Introducing Bias

Semantic Objects: Features

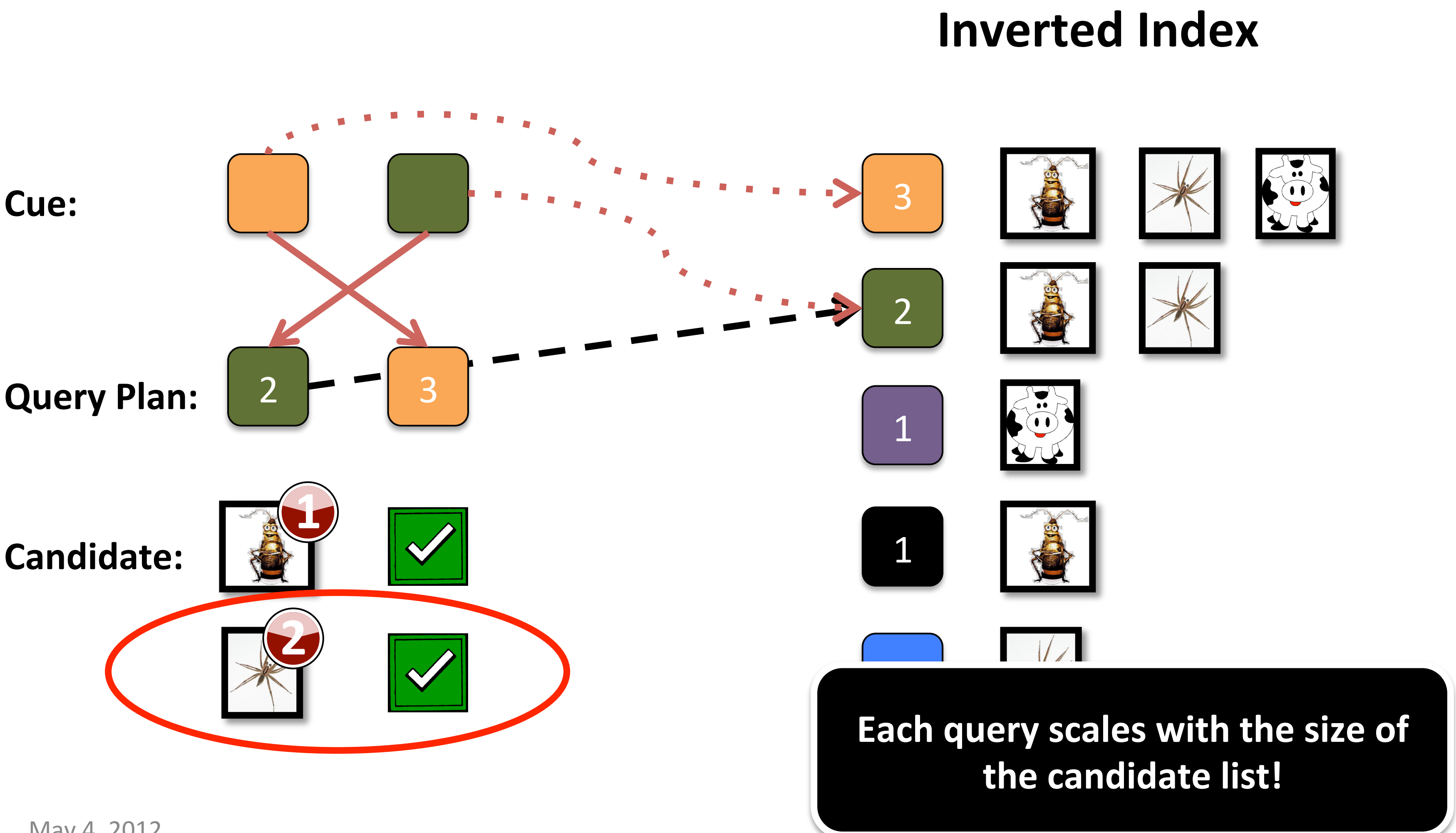


Inverted Index



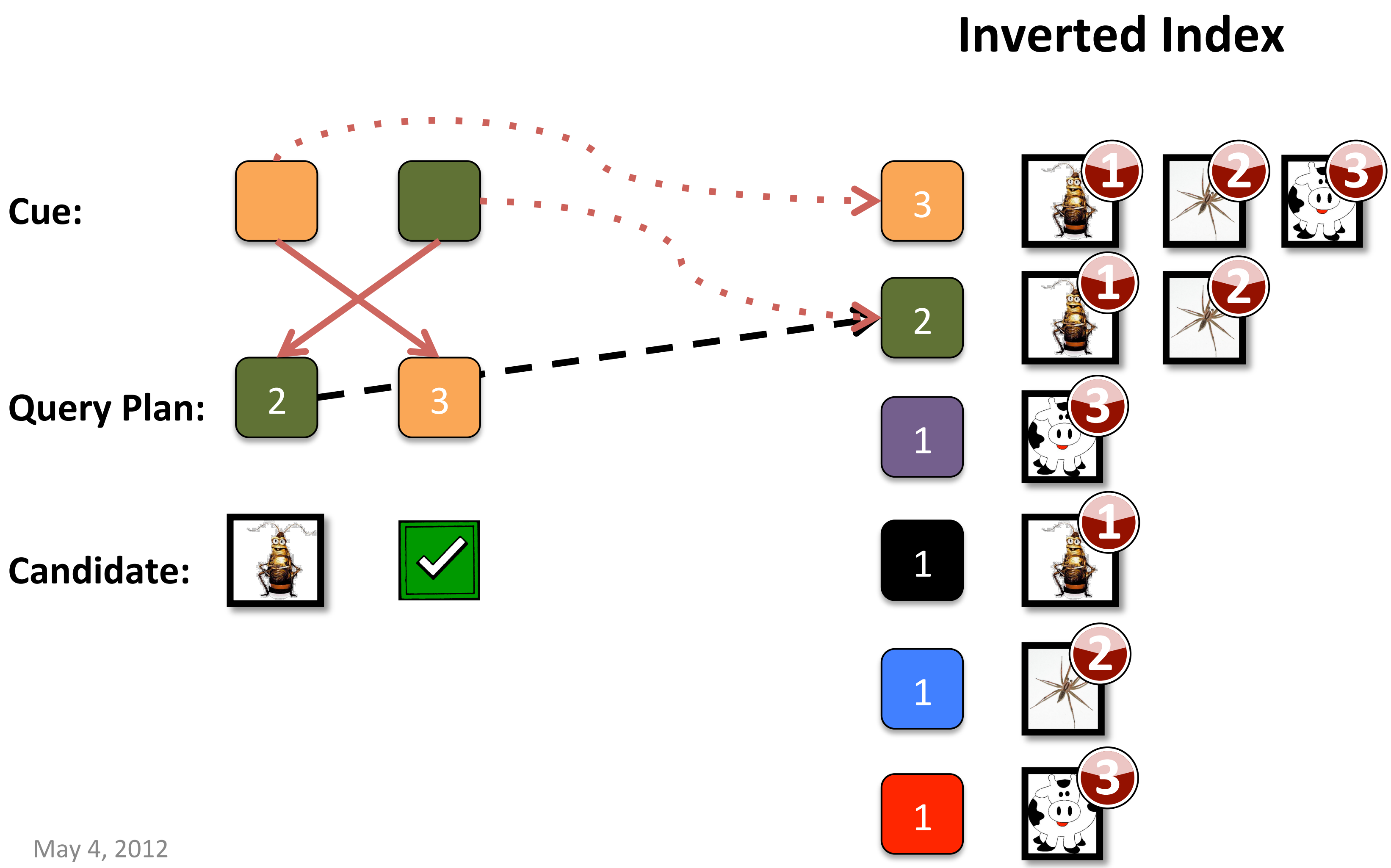
Biased Retrieval Algorithm #1

Sort on Query



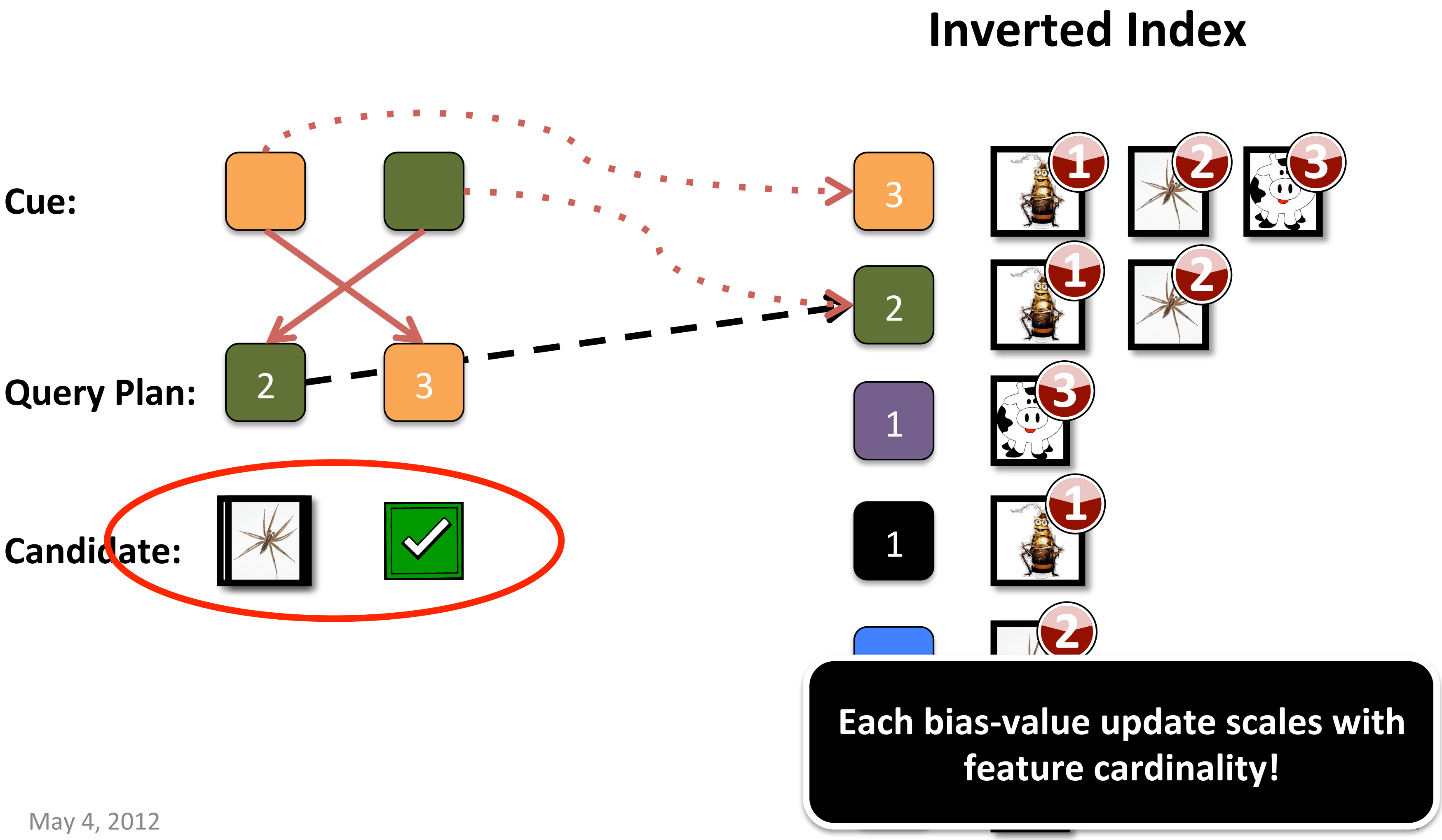
Biased Retrieval Algorithm #2

Static Sort



Biased Retrieval Algorithm #2

Static Sort



Our Hybrid Approach

Empirically supported cardinality threshold, θ

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

- Candidate enumeration scales with # of objects with large cardinality (which *should* be rare)

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

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

Empirical Evaluation
[ICCM '10], [AAAI '11]

Performance Characterization

- Selectivity + Co-occurrence

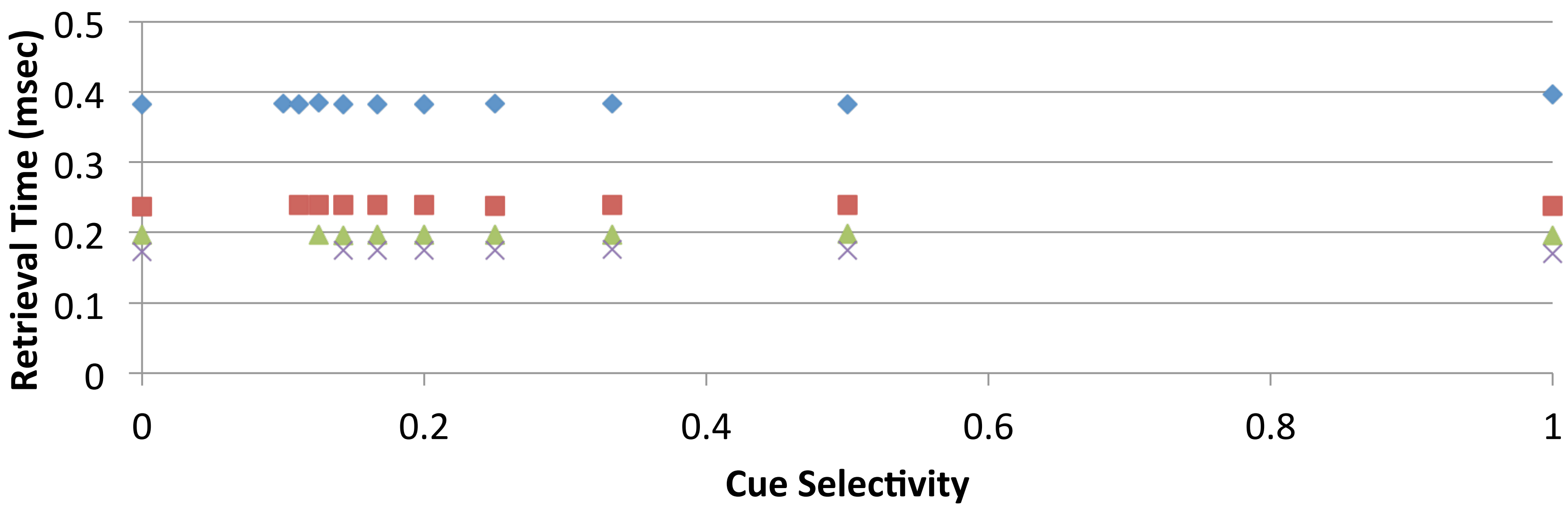
  O(Failed Candidates)

Empirical Evaluation

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

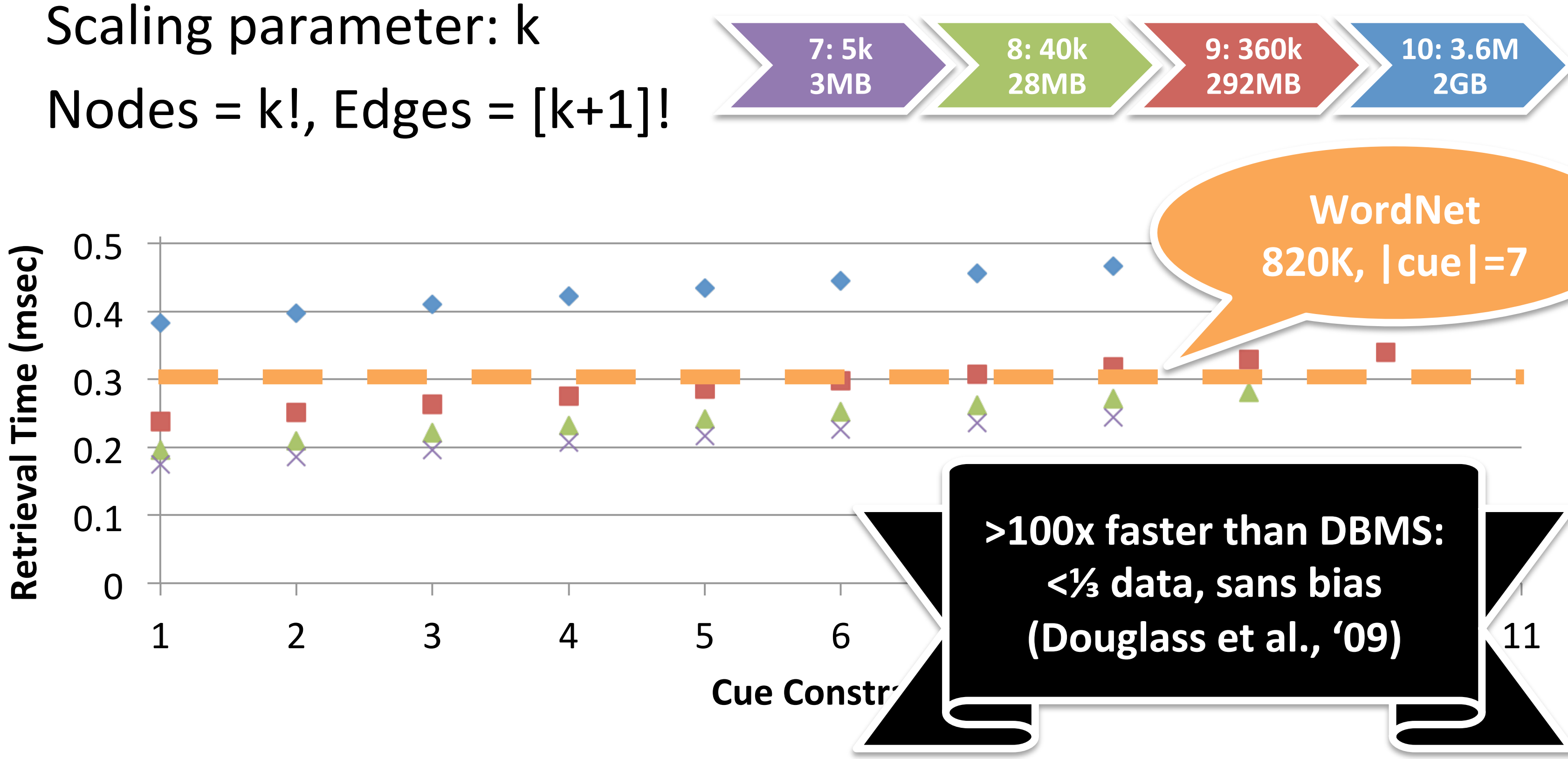
Synthetic Evaluation

- Scaling parameter: k
- Nodes = k!, Edges = [k+1]!



Synthetic Evaluation

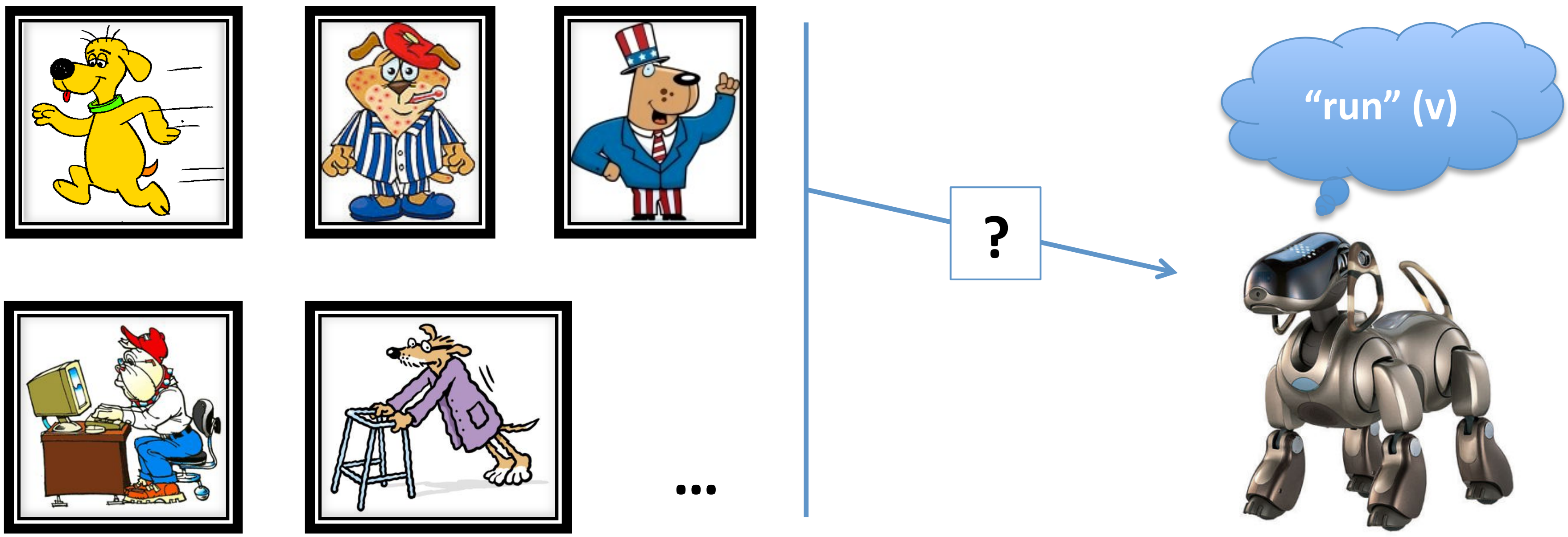
- Scaling parameter: k
- Nodes = k!, Edges = [k+1]!



WSD Evaluation

Motivation

Memory **Agent**

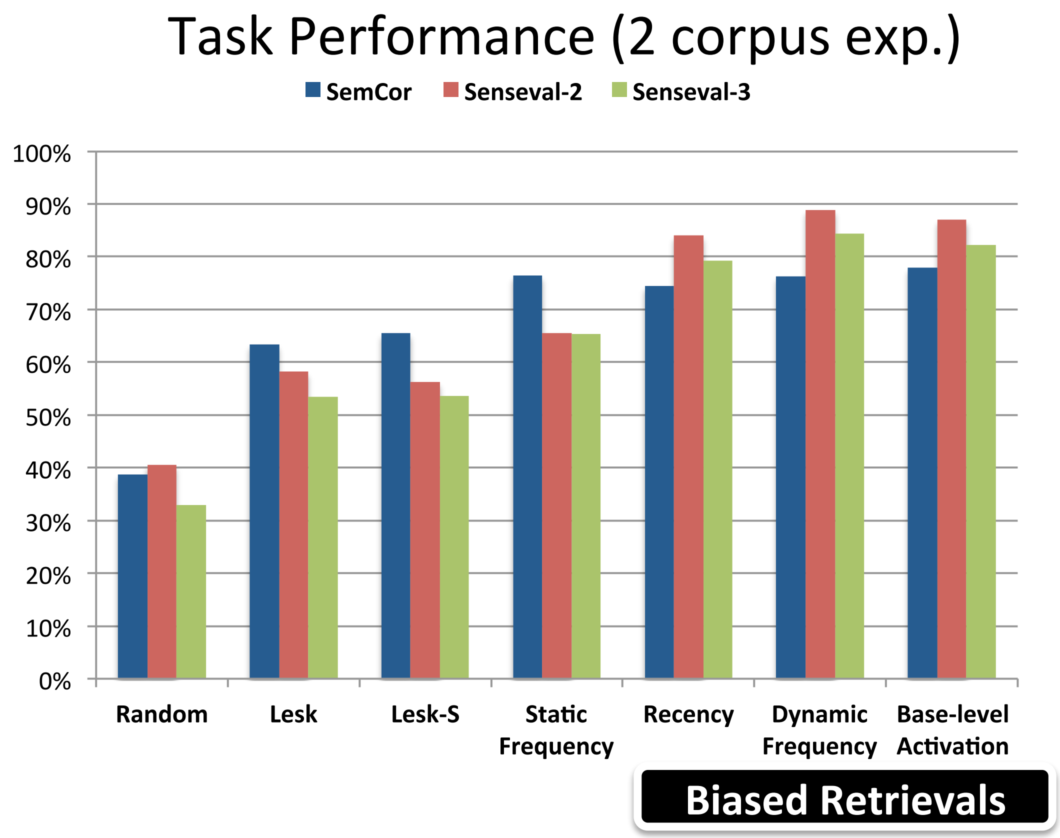


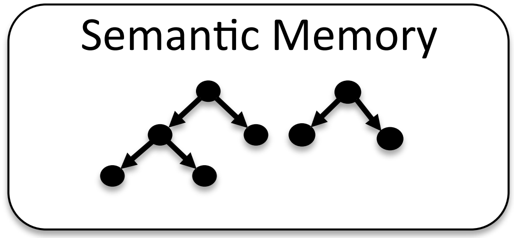
Problem. Ambiguous Cues
Hypothesis. Retrieval History is Useful
Application. Word Sense Disambiguation

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*(# obj's), ≤ 13.25 msec.
  - *Locally Efficient* Approximation: *O*(1), ≤ 1.34 msec.





- 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

Forgetting

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.

Approach

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

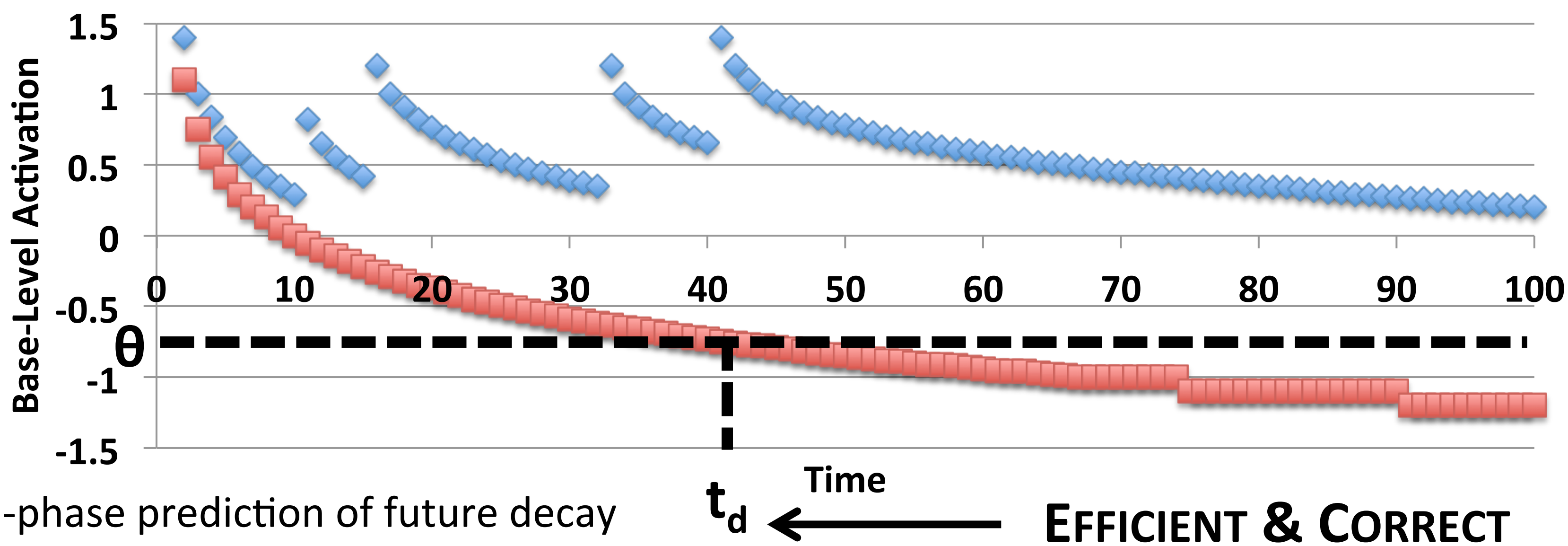
Task Independent

Base-Level Decay
(Anderson et al., 2004)

Predict future usage via history

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

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



2-phase prediction of future decay

- Novel approximation
- Binary parameter search

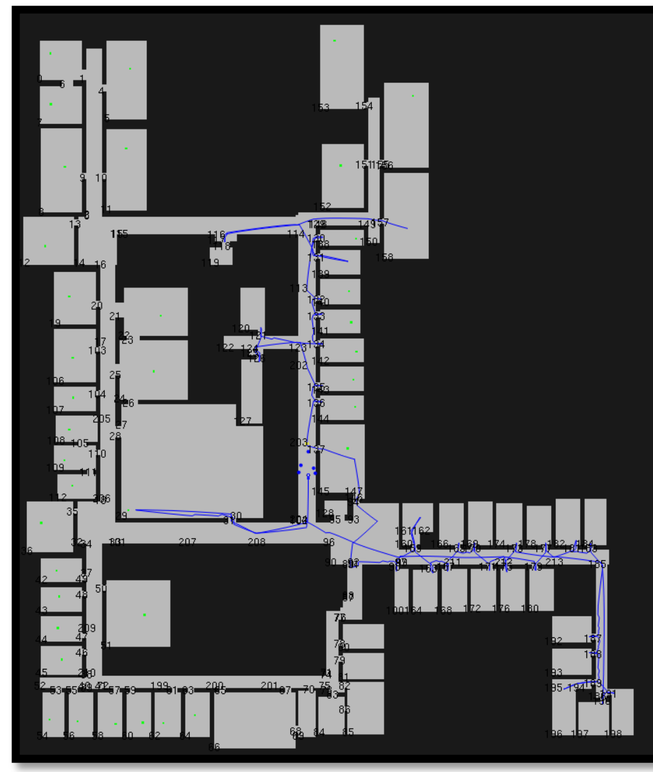
EFFICIENT & CORRECT

~~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

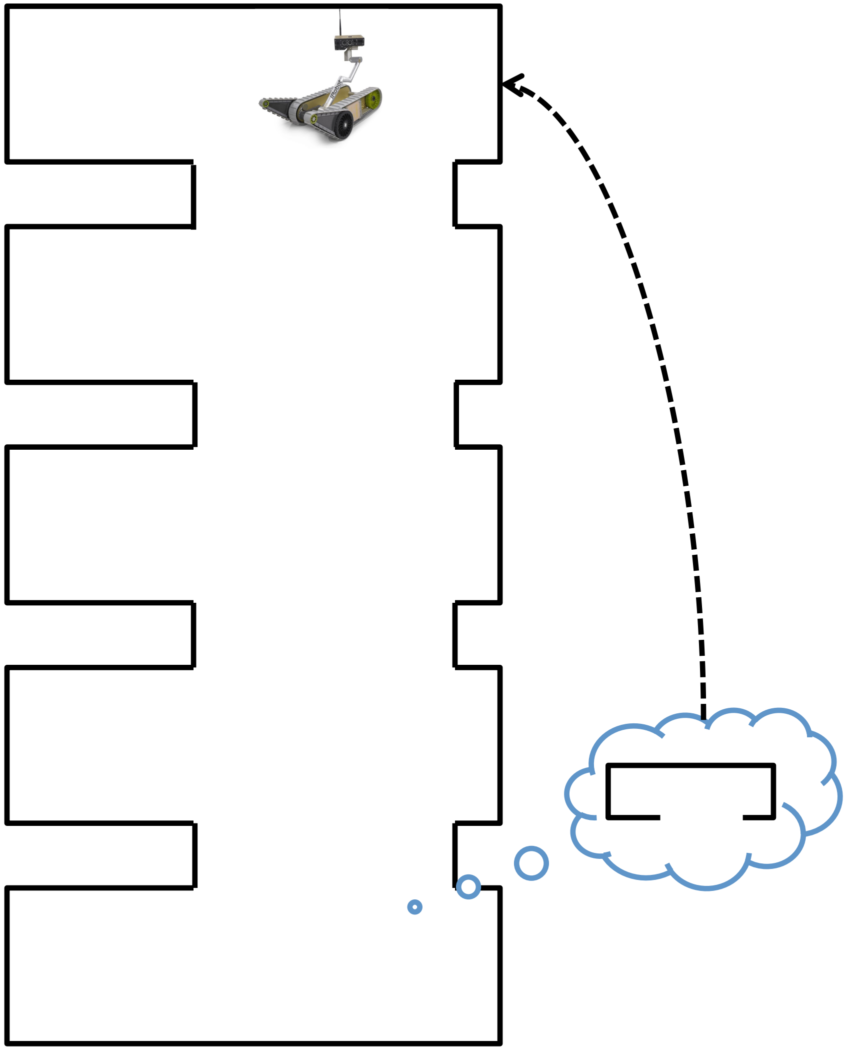
- Minor: rule matching (Forgy, 1982)
- Major: episodic reconstruction
  - |episode|~|working memory|

Forgetting Policy. Memory hierarchy

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

Task Independent

Map Knowledge



Room Features

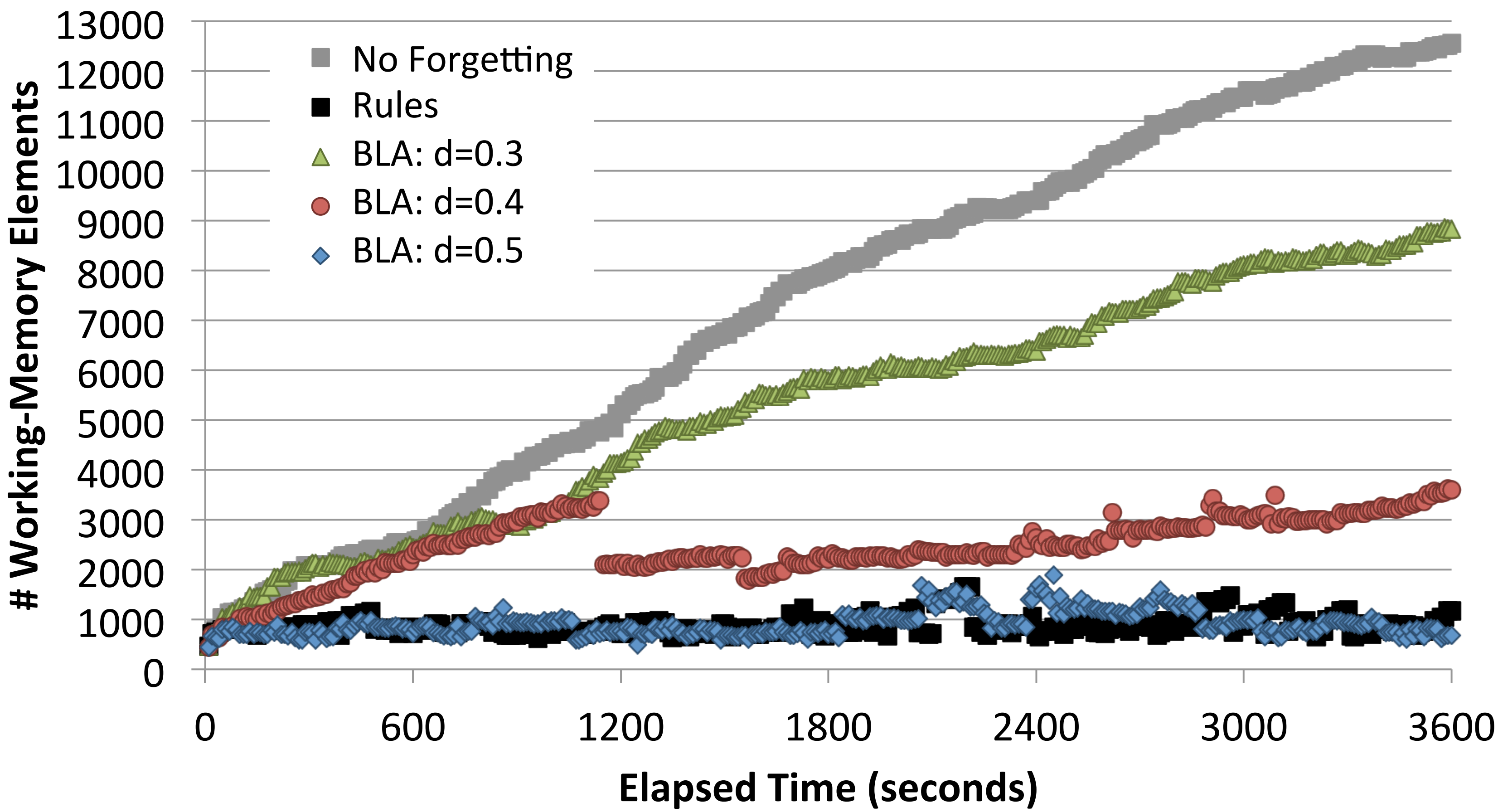
- Position, size
- Walls, doorways
- Objects
- Waypoints

Usage

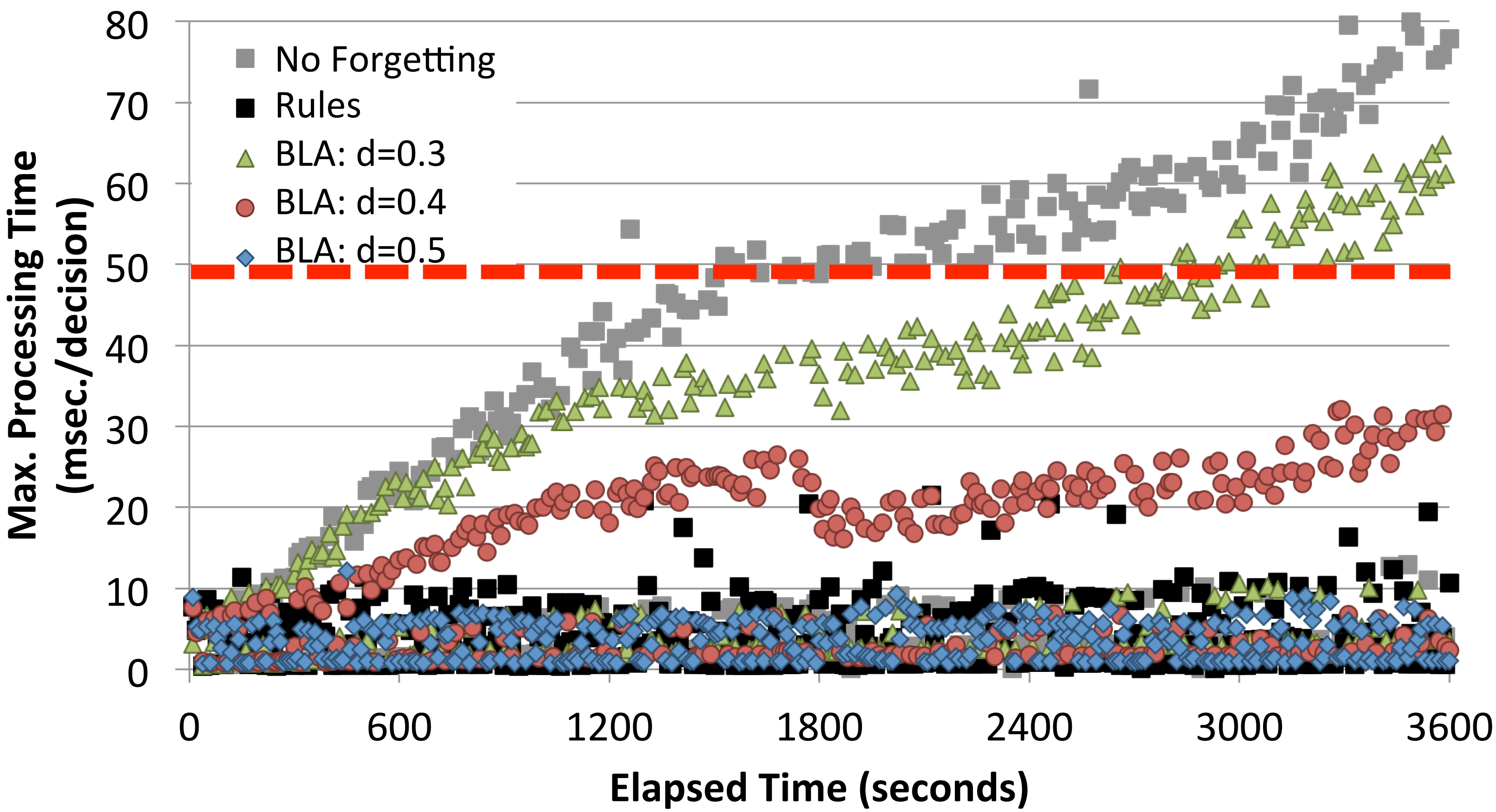
- Exploration (-->SMem)
- Planning/navigation (<--SMem)

  Reconstruction

Results: Working-Memory Size



Results: Decision Time

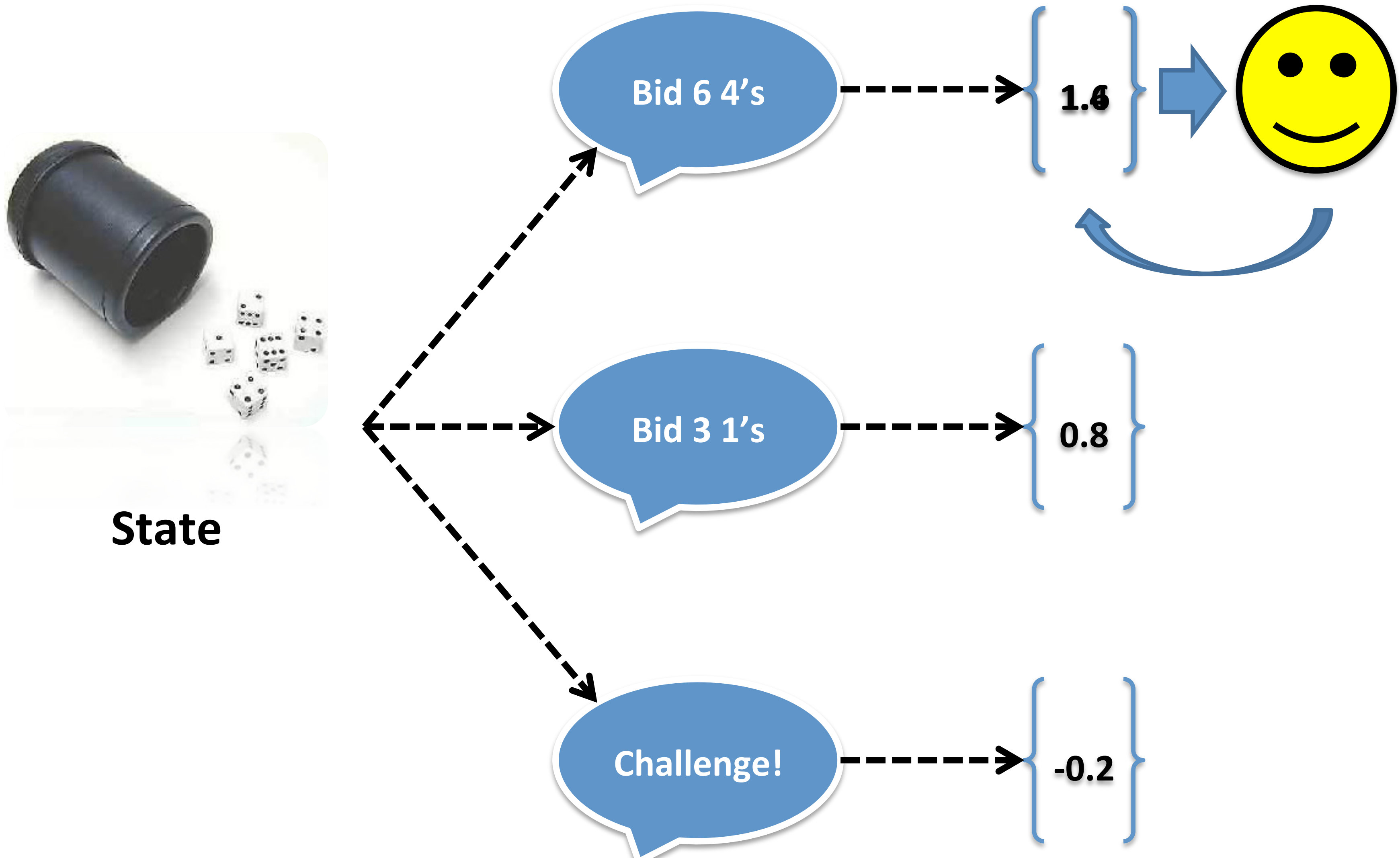


Task #2: Liar's Dice

- 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



Problem: Memory Consumption

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

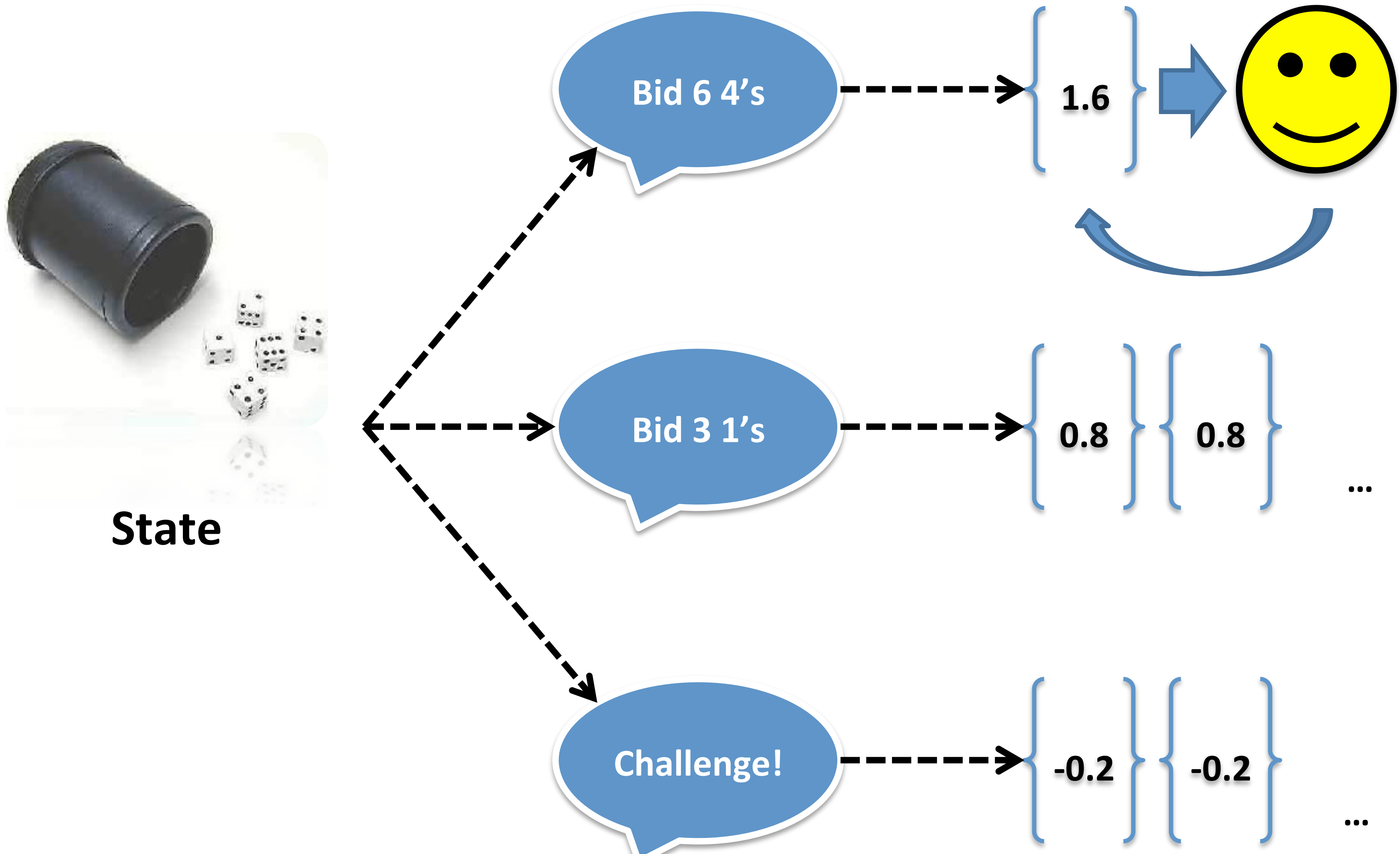
- Soar: procedural knowledge (*RL* rules)
- Many possible actions per turn;
  at most feedback for a single action

Forgetting Policy. Keep what you can't reconstruct

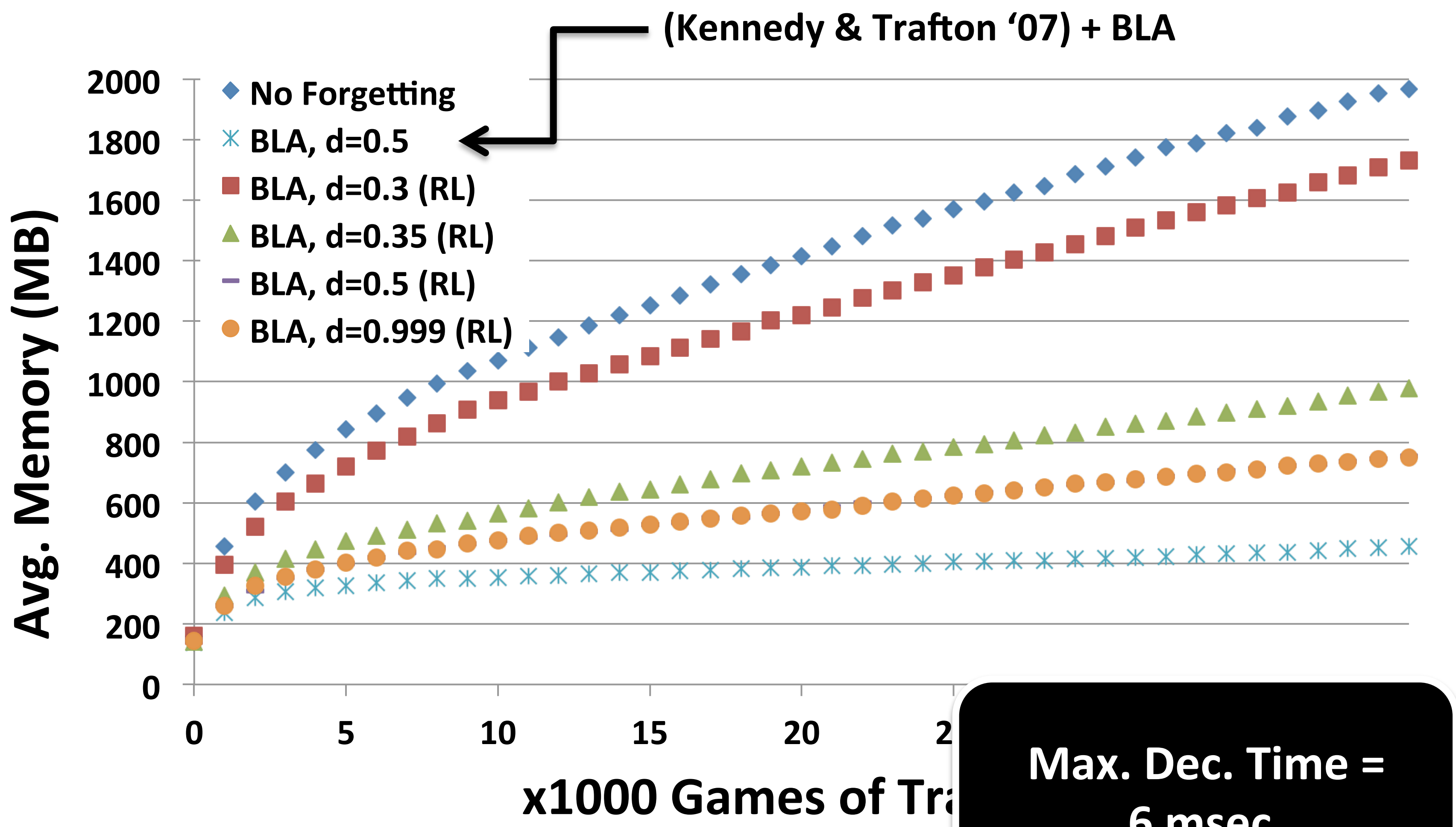
1. Forget unused RL rules that have not been rewarded
2. Learn rules via reasoning as necessary ("chunking")

Task Independent

Forgetting Action Knowledge

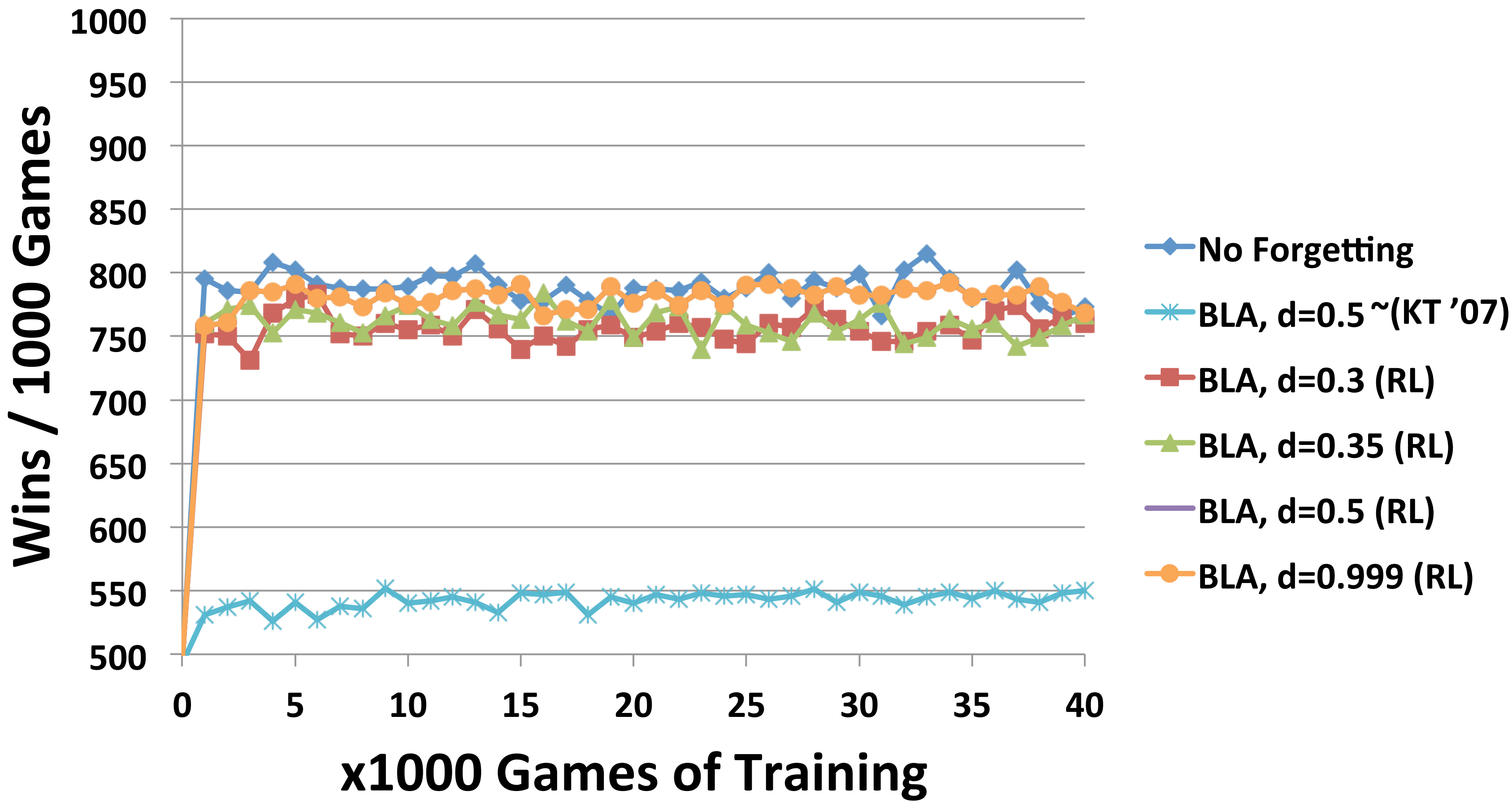


Results: Memory Usage

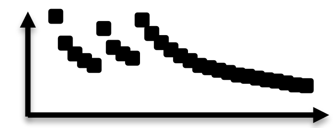


Max. Dec. Time =
6 msec.

Results: Competence



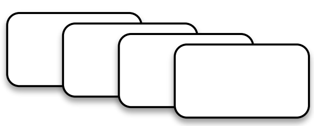
Forgetting



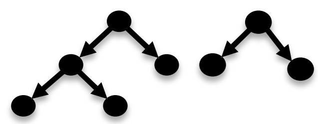
- Explored common forgetting hypothesis in two memories, two complex tasks
- Developed <u>efficient</u> and <u>correct</u> method of forgetting via base-level activation model
- Improves <u>reactivity</u> and <u>scaling</u> for long lifetimes and large amounts of knowledge, with <u>high task performance</u>

Summary of Dissertation

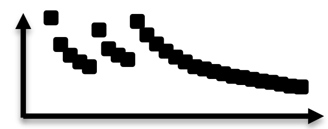
Episodic Memory

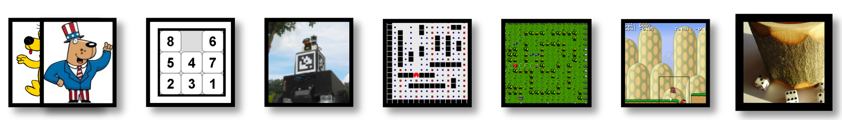


Semantic Memory



Forgetting



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

Discussion of Contributions
Soar

- Extends agents
  - Capabilities: more comprehensive access to experience -> more complex tasks
  - Lifetime: can work on time scales never attempted before
- Active use at UM, SoarTech, ICT, BYU, UP, …
- Basis for further research into memory functionality, integration, and use

Discussion of Contributions

Cognitive Architecture & Modeling

- Radically extends the complexity and lifetime of behavior that can be modeled
- Depth and breadth of evaluation serve as benchmark for research into general memory mechanisms
- Deepen and broaden understanding of general environment/task/agent properties and task/architecture-independent knowledge representations/processes

Discussion of Contributions
Knowledge/Agent-based Systems

- Analyses and algorithms inform integration of history and reasoning
- General approach for scaling RL to large state spaces

Questions?

Thank You :)

JOHN, COMMITTEE MEMBERS, SOAR

MELANIE H, ANDREA A, KYLA M, JIN H, GARGI D, LAURA F, MARK H, JIM B, OLGA K, KAUSHIK V, KELLY C, KAREN A, LUCIE H, DAVE C, SUSAN M, GEORG E, RADA C, FRANCINE D, JULIE W