

Episodic Memory in Soar

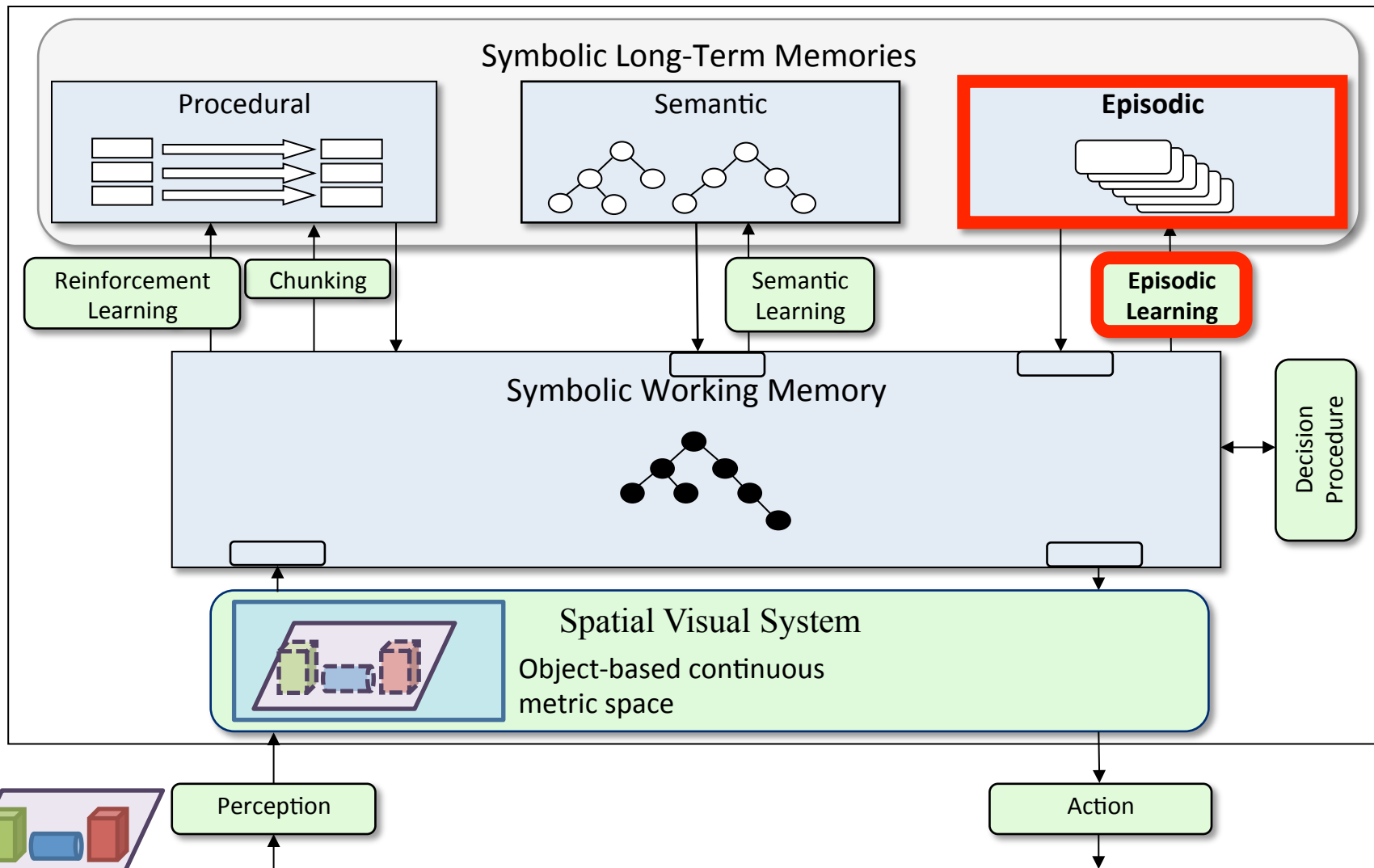
Nate Derbinsky

Wentworth Institute of Technology

Agenda

- Episodic memory as a learning mechanism
- Basic usage
- Multi-domain scaling evaluation

Soar 9



Episodic Memory

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



Episodic Memory

Integration

Representation

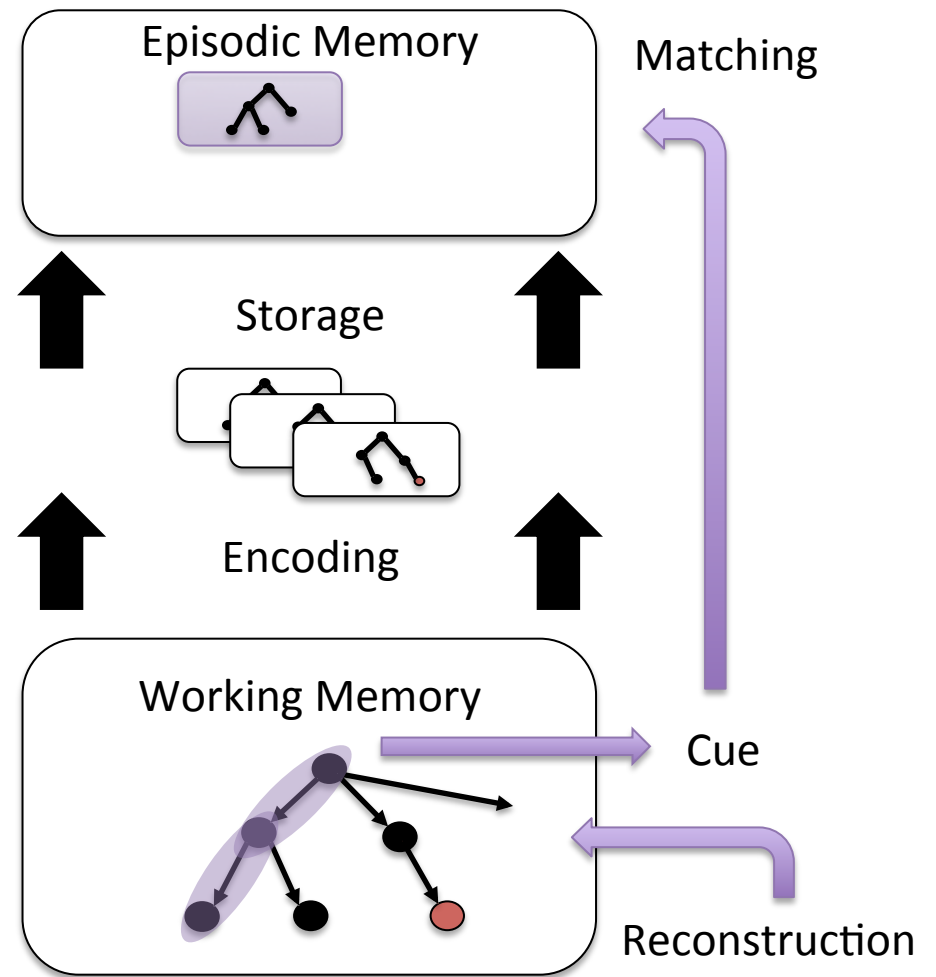
- Episode: connected di-graph
- Store: temporal sequence

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



Episodic Memory in Soar

A weak learning mechanism

- Automatically captures, stores, and temporally indexes agent state
- Provides content-addressable agent interface to autobiographical prior experience
- Supports a general set of cognitive capabilities

Basic Usage

- Working-memory structure
- Episodic-memory representation
- Storing knowledge
- Retrieving knowledge

Working-Memory Structure

Soar creates an **epmem** structure on each state

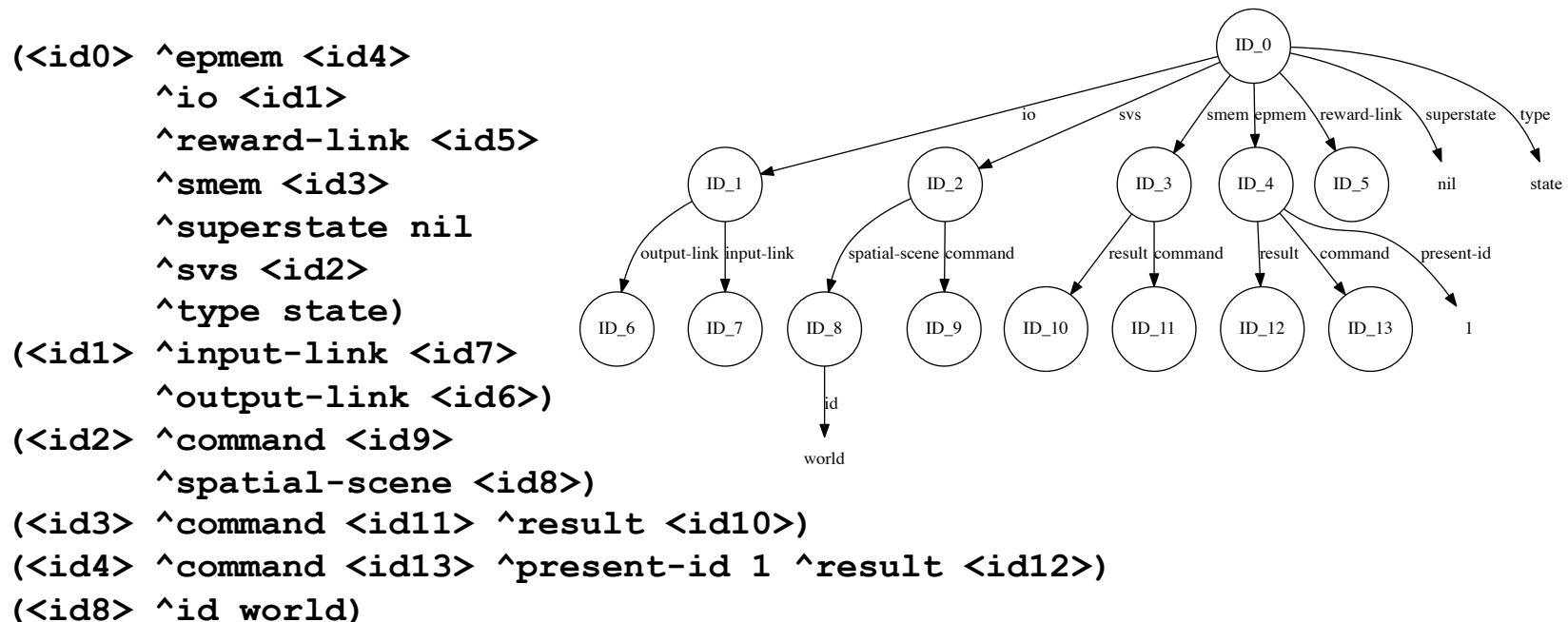
Each **epmem** structure has specialized substructure

- **command**: agent-initiated actions
- **result**: architectural feedback
- **present-id**: current episode number (more later)

Episodic-Memory Representation

Similar to working memory: symbolic triples

- Structures within an episode are connected; separate episodes are disconnected



Storing Knowledge

- **What.** top state of working memory
- **Why.** task-independent trigger
 - `epmem --set trigger << dc output >>`
 - `dc`: decision cycle (default)
 - `output`: new augmentation of output-link
- **When.** at the end of a phase
 - `epmem --set phase << output selection >>`
 - `output` is default
 - `selection` may be useful for in-the-head agents

Retrieving Knowledge

Cue-Based

Find the episode that best matches a cue and add it to working memory

Temporal Progression

Replace the currently retrieved episode with the next/previously encoded episode

Non-Cue-Based (not covered)

Add an episode to working memory from episode #

Common Constraints (motivated by performance/reactivity):

- Only one per state per decision
- Processed during `phase` (slide 10)
- Only re-processed if WM changes to commands

Cue-Based Retrieval: Syntax

(**<epmem>** ^**command** **<cmd>**)

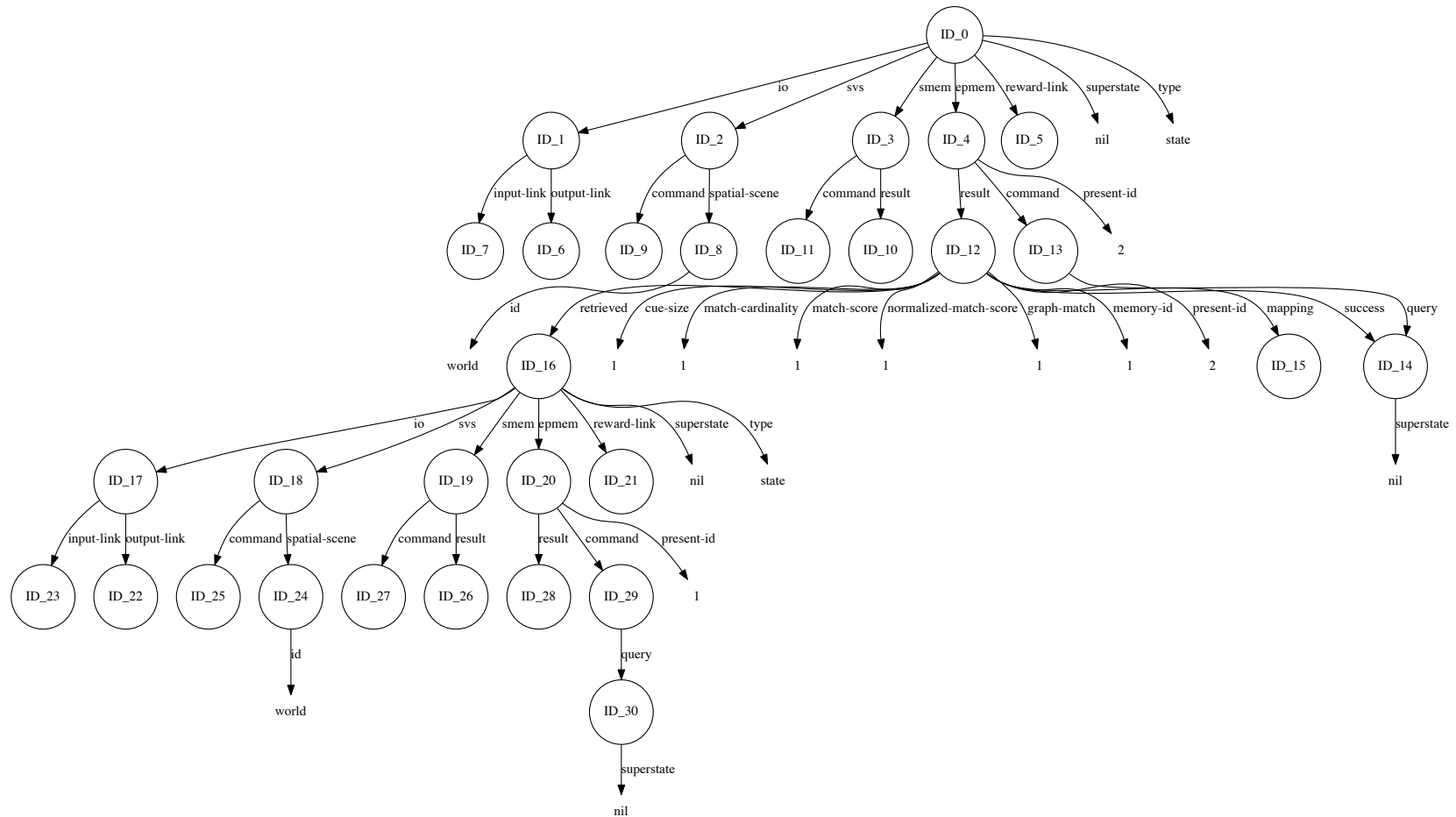
(**<cmd>** ^**query** **<q>**

^**neg-query** **<nq>**)

- The **neg-query** is optional
- Cues must be acyclic
- The **<q>** and **<nq>** identifiers form the roots of episode sub-graph cues
 - **query** represents desired structures
 - **neg-query** represents undesired structures

Cue-Based Retrieval: Example

Result



Cue-Based Retrieval

Optional Modifiers

`(<cmd> ^before time-id)`

`(<cmd> ^after time-id)`

`(<cmd> ^prohibit time-id1 time-id2 ...)`

Hard constraints on the episodes that can be retrieved.

Temporal Progression

`(<cmd> ^next <new-id>)`

`(<cmd> ^previous <new-id>)`

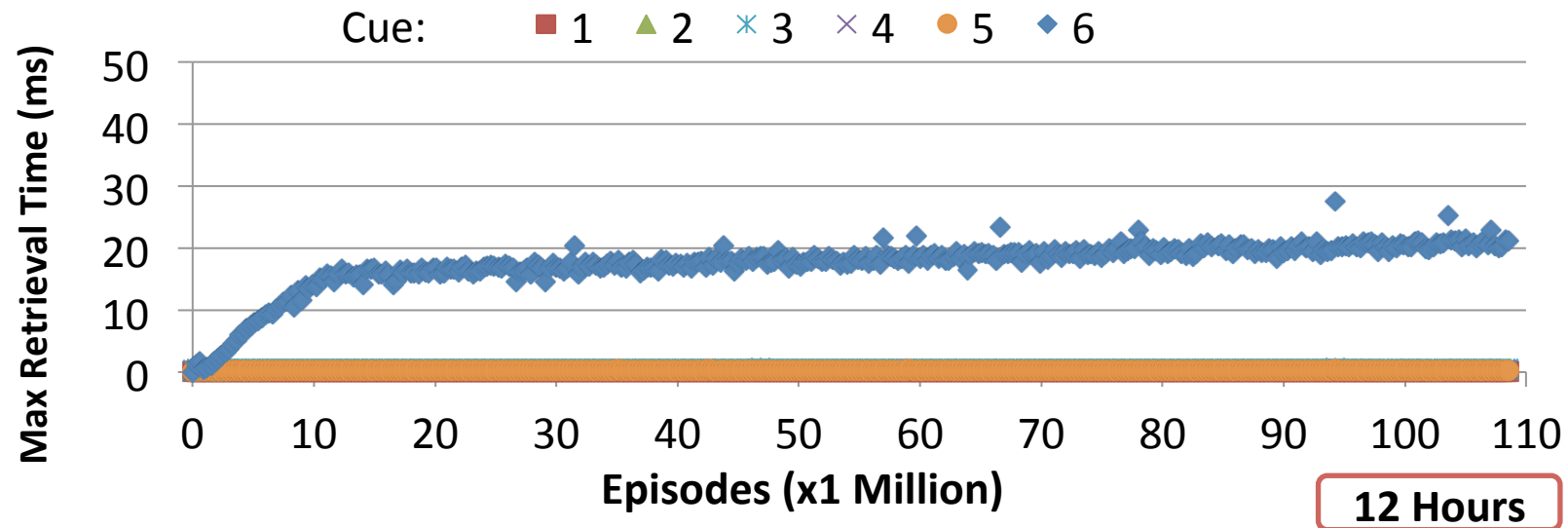
Retrieves the next/previous episode,
temporally, with respect to the last that was
retrieved

Multi-Domain Scaling Evaluation

Experimental Setup



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



Thank You :)

Questions?