Oscillatory Neural Network Models of Sequential Short-Term Memory

June 15, 2010

Jared Sylvester
Comp. Sci.

Scott Weems
CASL, Psychology

Jim Reggia
Comp. Sci., CASL

with Mike Bunting

Sylvester J, Reggia J, Weems S and Bunting M. “A temporally asymmetric Hebbian network for sequential working memory.” Int’l Conf. on Cognitive Modeling, August 2010. In press.

Short-Term Memory

short-term memory refers to the human memory system that retains information over brief time intervals (on the order of seconds)

characterized by substantial *capacity limitations* in contrast to the relatively limitless capacity of more permanent long-term memory:

approximately four items [Cowan et al, 2005]

Conceptual overview

Problem

Need to improve foreign language comprehension

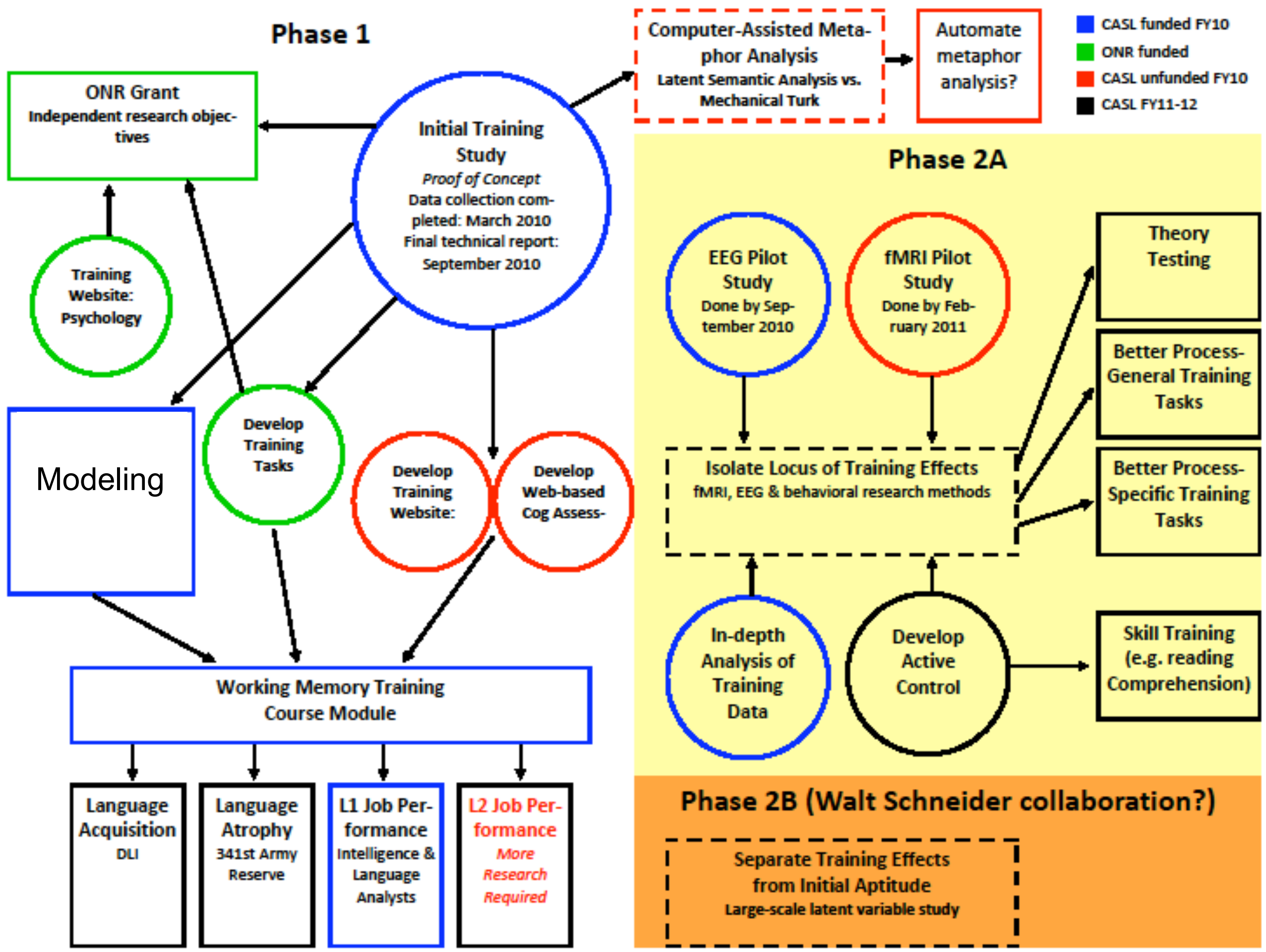
Relevance

Working memory is critical for comprehension

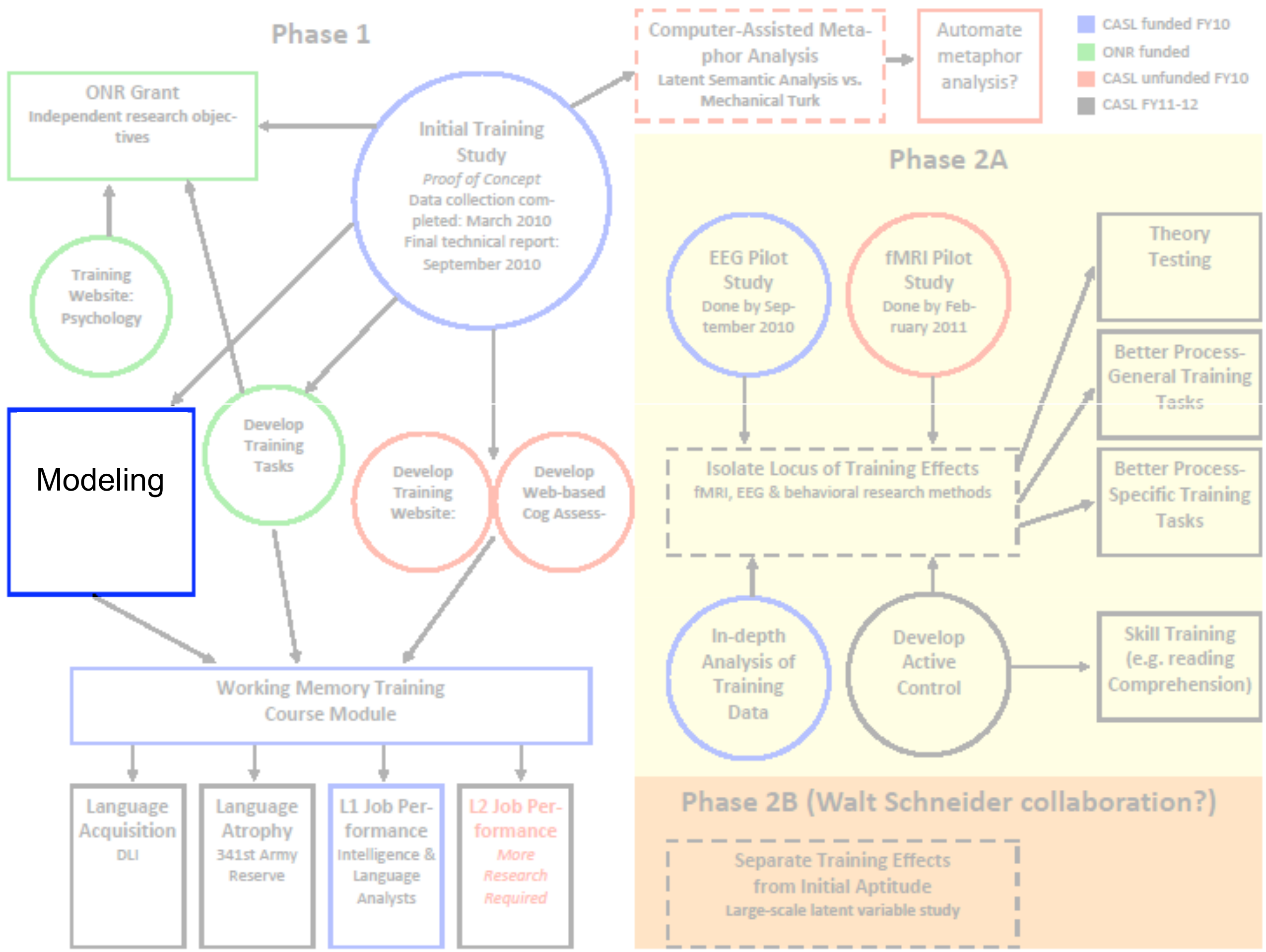
Goal

Improve comprehension through working memory training

TTO 3501: The Overview



TTO 3501: The Overview



Goals for the Computational Modeling

To identify individual difference variables that predict training benefits

1. Develop machine learning/classifiers to make training effect predictions

To explore how items are retained in working memory to better understand what the training is changing

2. Develop simple attractor models of short term memory simulating human performance
3. Expand those models to include cognitive control elements

Goals for the Computational Modeling

To identify individual difference variables that predict training benefits

1. Develop machine learning/classifiers to make training effect predictions

To explore how items are retained in working memory to better understand what the training is changing

2. Develop simple attractor models of short term memory simulating human performance
3. Expand those models to include cognitive control elements

Neural Modeling: Goal

- Examine the relative roles of decay and interference in determining short-term memory capacity

by

- Developing *simple oscillatory models* of short-term memory with decay
- comparing the models' performances to experimental results from human subjects

Overview

- Background:
  - behavioral data
  - neural models of memory
    - fixed attractor networks
    - oscillatory networks
- Initial Model: Oscillatory Networks with Decay
  - model properties
  - comparison to behavioral data
- Updated Model: Temporally Asymmetric Weights
  - model properties
  - results

Overview

- Background:
    - **behavioral data**
    - neural models of memory
        - fixed attractor networks
        - oscillatory networks
- Initial Model: Oscillatory Networks with Decay
    - model properties
    - comparison to behavioral data
- Updated Model: Temporally Asymmetric Weights
    - model properties
    - results

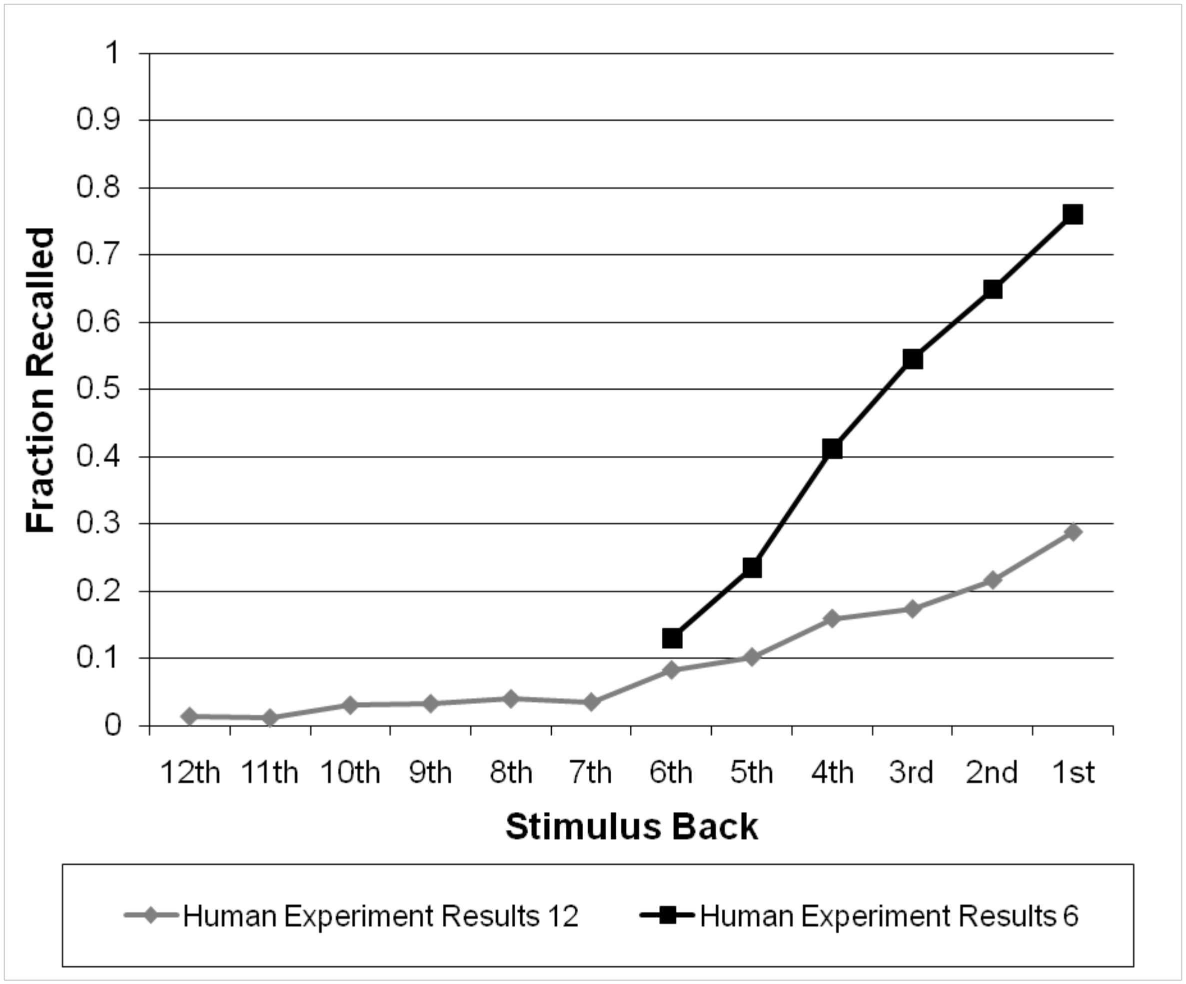
The View from Experimental Psychology

Behavioral Task: Running Memory Span

- 38 adult subjects; part of a larger study
- sequence of digits presented in rapid succession
  - 1 3 6 8 4 5 9 2 … (12 to 20 digits)
  - 2 per second presentation rate
  - Not aware of when sequence will end
- subject expected to retain and repeat the most recently seen 6 or 12 digits of the sequence
  - digits entered by mouse clicks
- accuracy:
  - number correctly recalled in correct position
  - results averaged over twelve trials per subject

The View from Experimental Psychology

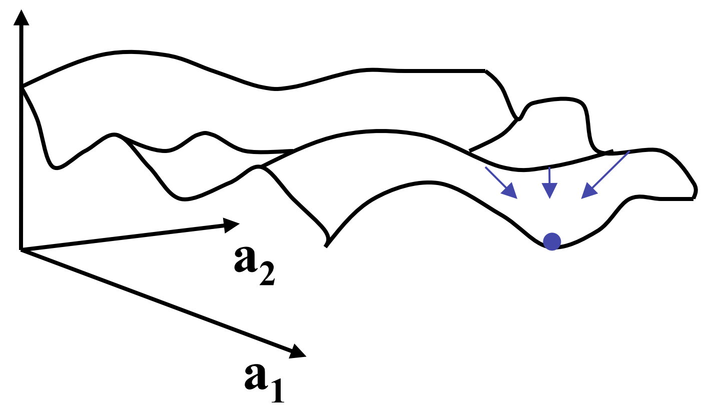
Behavioral Task



Overview

- Background:
  - behavioral data
  - **neural models of memory**
    - fixed attractor networks
    - oscillatory networks
- Initial Model: Oscillatory Networks with Decay
  - model properties
  - comparison to behavioral data
- Updated Model: Temporally Asymmetric Weights
  - model properties
  - results

Attractor Neural Network Models of Memory



STM = activity patterns

LTM = connection weights *W*

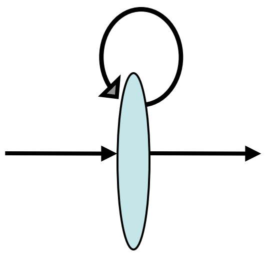
partial, noisy, or random pattern → network → restored pattern

Content-addressable memory:

- stored memory is attractor state of network
- usually involves fixed point attractors
- however, growing interest in oscillatory attractors
  - brain highly oscillatory
  - multiple patterns active

Fixed-Point Attractor Networks

Hopfield networks, brain-state-in-a-box, and related models



- recurrent network structure
- content-addressable memory system
- store memory patterns by changing the weights w_{ij} on connections between nodes
- *Hebbian learning* used
  - strengthen connections between co-active nodes

Storage: Hebbian Learning

- Network: *N* nodes, fully connected

  node activity $a_i = \pm 1$

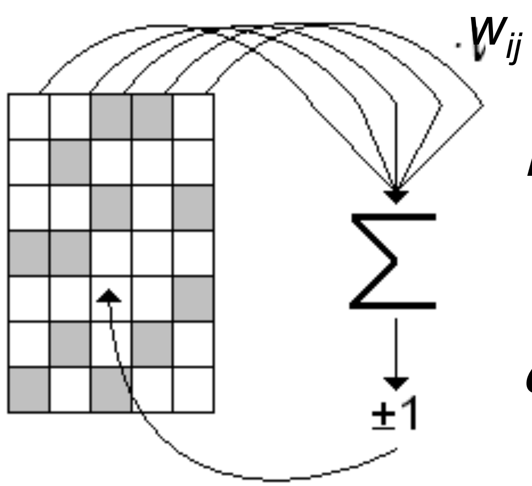
- Memories: $\vec{a}^{\,p}, \quad p = 1, 2, \ldots, M$

- Storage: $$w_{ij} = w_{ij}^{old} + \frac{1}{N} a_i a_j$$ (except $w_{ii} = 0$)

Memory storage is order independent!

Recall

- **Recall**: randomly and asynchronously do the following



$$h_i(t) = \sum_j w_{ij} a_j(t) - \theta_i$$

$$a_i(t+1) = \pm 1 \text{ with probability } \left(1 + e^{\mp 2h_i/T}\right)^{-1}$$

- traditionally terminates when there is no longer any change in the network state
- cause of failure to recall stored stimuli: interference

Simple Oscillatory Networks

• uses same method for storing patterns

- node thresholds θ_i change to induce oscillation
- initially during recall $\theta_i = 0$
- when $a_i = +1 \rightarrow \theta_i$ rises
- when $a_i = -1 \rightarrow \theta_i$ drops
- these changes make it harder as time passes for a node to remain in a single state

Threshold Dynamics:

$$\theta_i(t) = bR_i(t)$$

$$R_i(t+1) = \frac{R_i(t)}{c} + a_i(t+1)$$

$$b = 0.15, c = 1.2$$

$\Rightarrow$ network oscillates between stored memory patterns

Overview

- Background:
  - behavioral data
  - neural models of memory
    - fixed attractor networks
    - oscillatory networks
- Initial Model: Oscillatory Networks with Decay
  - model properties
  - comparison to behavioral data
- Updated Model: Temporally Asymmetric Weights
  - model properties
  - results

Simple Oscillatory Model of STM

- the weights now stores memories that decay
- **Storage**: $w_{ij} = (1 - k_d) w_{ij}^{old} + \frac{1}{N} a_i a_j$ (except $w_{ii} = 0$)

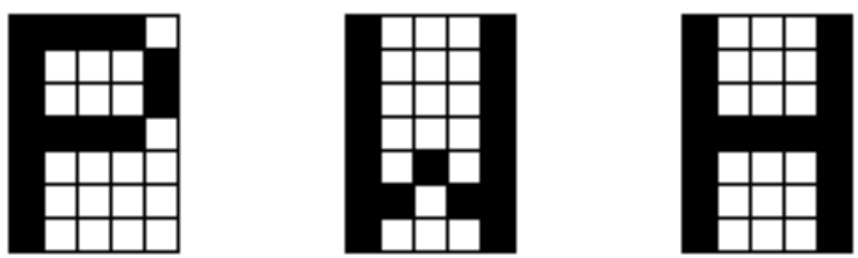
  where k_d is the *decay rate*

- **Recall**: same as before

  - memory storage is no longer order independent!
  - causes of failure to recall stored stimuli: interference *and* decay

Stimuli

- network structure
  - N = 35 nodes (7 rows and 5 columns)
- “arbitrary” stimuli used as patterns to be stored
- represented as letters for easy identification
  - letters A – Z represented as ±1 patterns
- examples:



■ = +1
□ = -1

Measuring Model's Retention of Stimuli

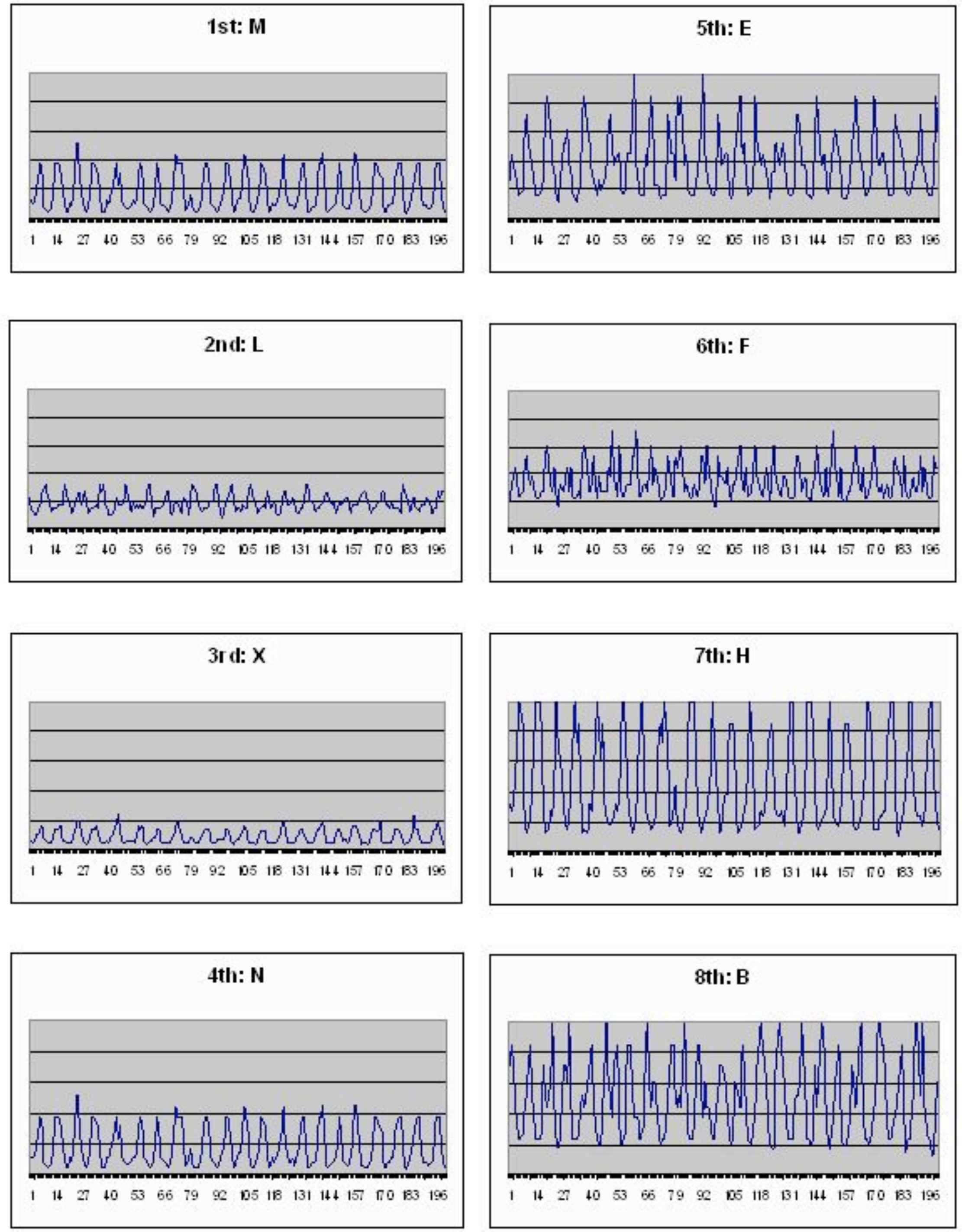
- train network on sequences of stimuli of different lengths
  - (4,8,12,16, or 20 - no repeated stimuli in a sequence)
- after storing each sequence:
  - start network in a random initial state of activity
  - run the network for 200 time steps as it oscillates
  - at each time step measure model's similarity to the memory patterns that served as stimuli
  - record all patterns that are perfectly remembered
- results averaged over hundreds/thousands of trials

Example

$k_d = 0.2$

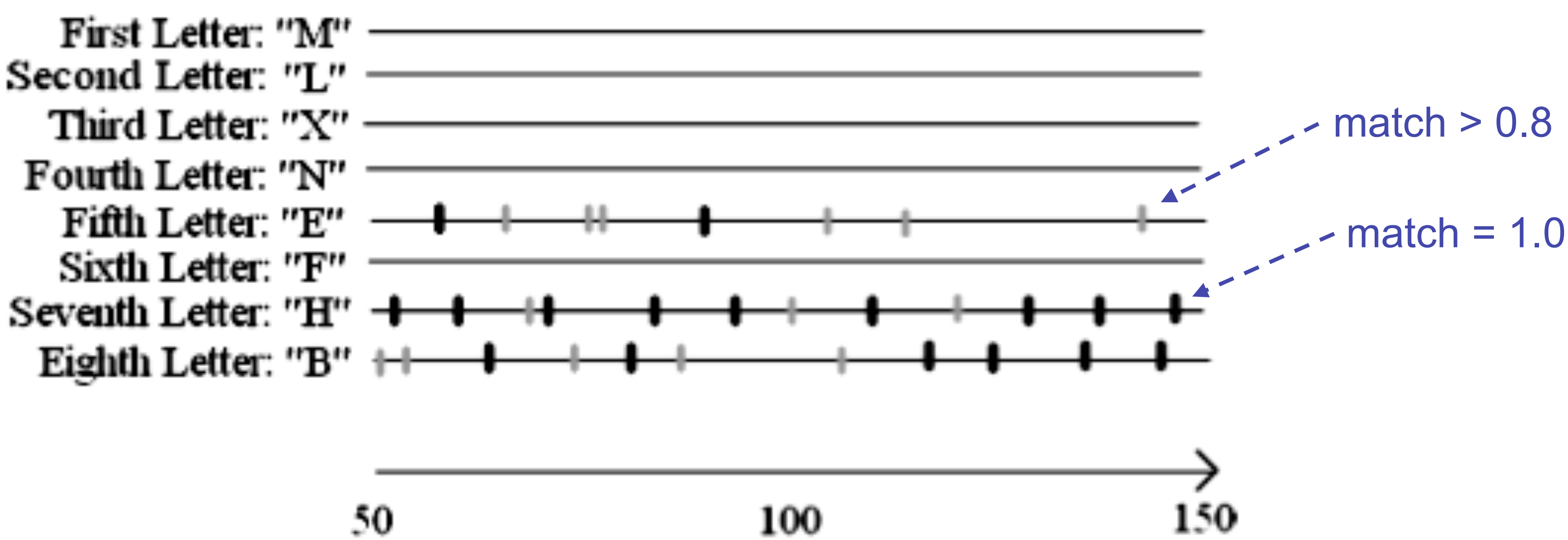
- 8 stimuli: MLXNEFHB
- random initial state
- more recently stored stimuli are more strongly recalled

Labeled as recalled:
E, H and B

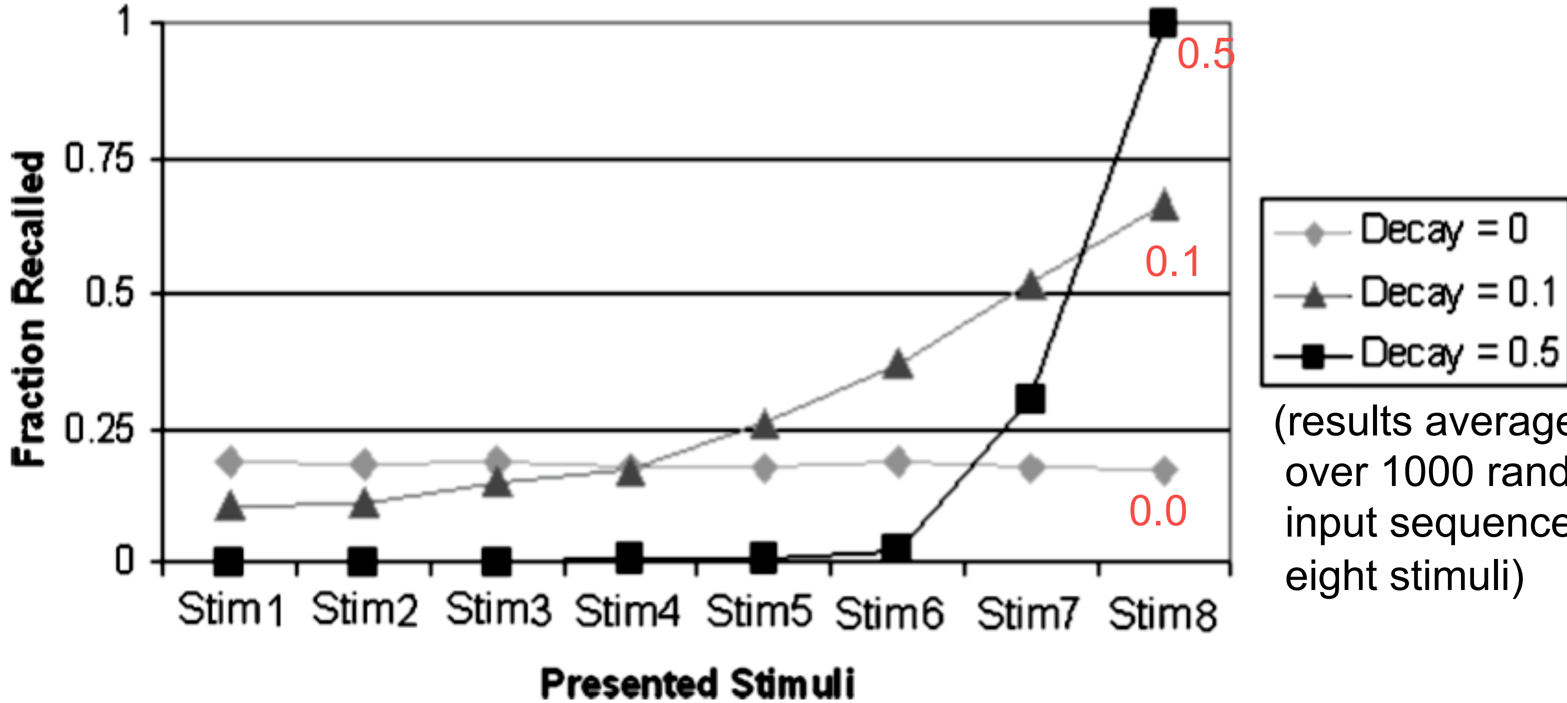


Same Example: Recall Behavior

- three perfectly recalled patterns
- alternating appearance in network



Fraction of Stimuli Recalled vs. Stimulus Position



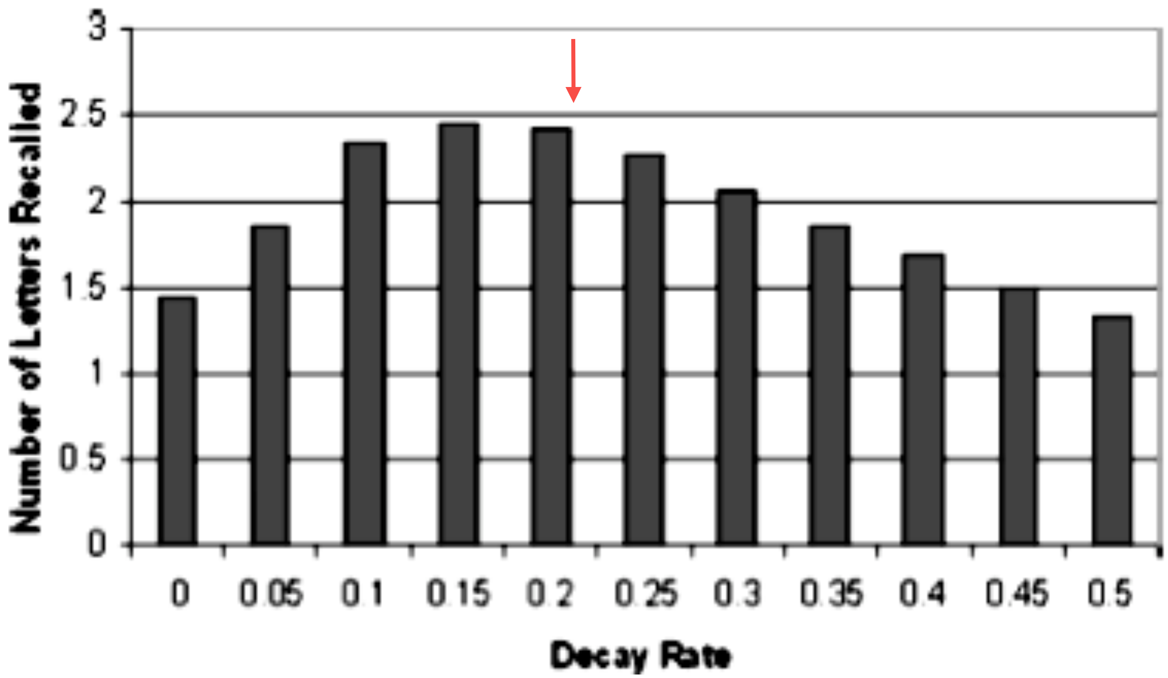
(results averaged over 1000 random input sequences of eight stimuli)

- *recency effect* observed with most decay rates k_d
- as decay increases
  - recent stimuli are more likely to be retained
  - older stimuli are less likely to be retained

Theoretical Analysis

- The non-linear, stochastic nature of model makes it difficult to analyze mathematically.
- What value of k_d will maximize the number of stimuli that are recalled?
- Analysis: both *interference* and *decay* contribute to losing stimuli from memory.
- Tradeoff: larger k_d leads to more decay but less interference.
- Result: intermediate values of k_d are optimal.
- For our stimuli, a very rough value of $k_d \approx 0.22$ is predicted to be optimal.

Stimuli Recalled with Different Decay Rates k_d

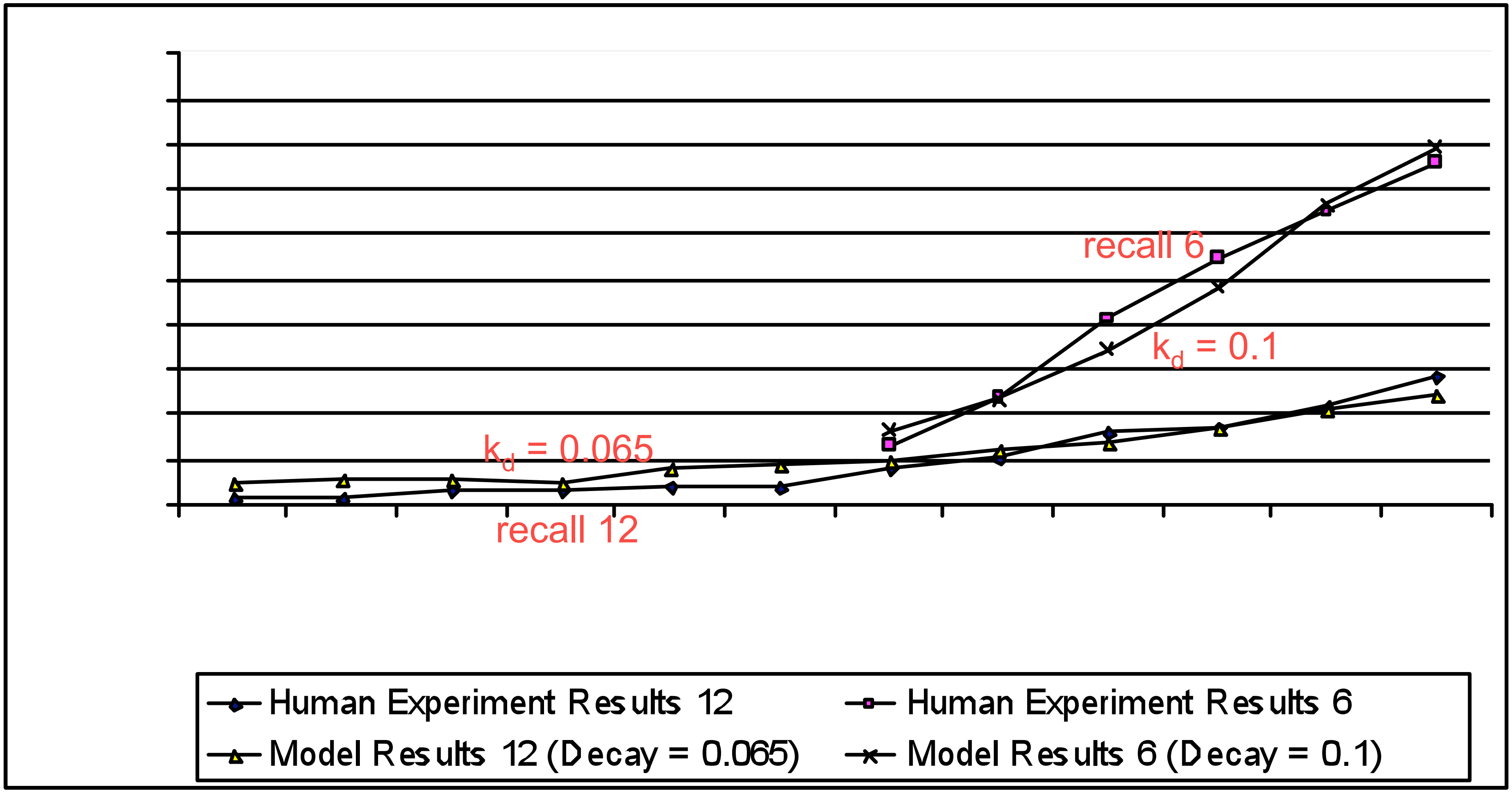


(results averaged over 1000 random input sequences)

less decay, **more interference**

more decay, less interference

Comparison of Model Results to Behavioral Data



Comments

- simple oscillatory model of short-term memory
- unlike past models, incorporates decay of stimuli
- demonstrates recency effect with non-zero decay rates
- both interference and decay prevent recall of stimuli
- can match behavioral data on different tasks simply by varying the decay rate used by the model
- hypothesis:
  Dynamic adjustments to activity decay rate may be an important aspect of the human attention mechanisms controlling forgetting.

Overview

- Background:
  - behavioral data
  - neural models of memory
    - fixed attractor networks
    - oscillatory networks
- Initial Model: Oscillatory Networks with Decay
  - model properties
  - comparison to behavioral data
- Updated Model: Temporally Asymmetric Weights
  - model properties
  - results

Revised Model: Temporally Asymmetric Weights

- previous model does not recall in order
  - human subjects do
  - rectify this discrepancy
- approach: add a second set of weights
  - similar, but *temporally asymmetric*
  - link node activity *now* with *previous* activity

Model Details

- N = 35 binary nodes
- W = N x N weight matrix
  - Same as before:

$$w_{ij}^{t} = (1 - k_d)w_{ij}^{t-1} + \frac{1}{N}a_i^t a_j^t (1 - \delta_{ij})$$

- V = N x N weight matrix
  - modified Hebbian learning
  - associates current state with previous state

$$v_{ij}^{t} = (1 - k_d)v_{ij}^{t-1} + \frac{1}{N}a_i^t a_j^{t-1}$$

Model Details

- modified input function
  - combine effects of *W* and *V*

$$h_i^t = \sum_j \left(\beta_1 w_{ij} a_j^t + \beta_2 v_{ij} a_j^{t-1}\right) - \theta_i^t$$

- other simplifications
  - non-probabilistic activation rule
  - simplified rules for θ

$$a_i^t = \begin{cases} +1 & h_i^t > 0 \\ a_i^{t-1} & h_i^t = 0 \\ -1 & h_i^t < 0 \end{cases}$$

Number of stimuli recalled

Model's answers are counted as recalled iff they are stored in the proper order

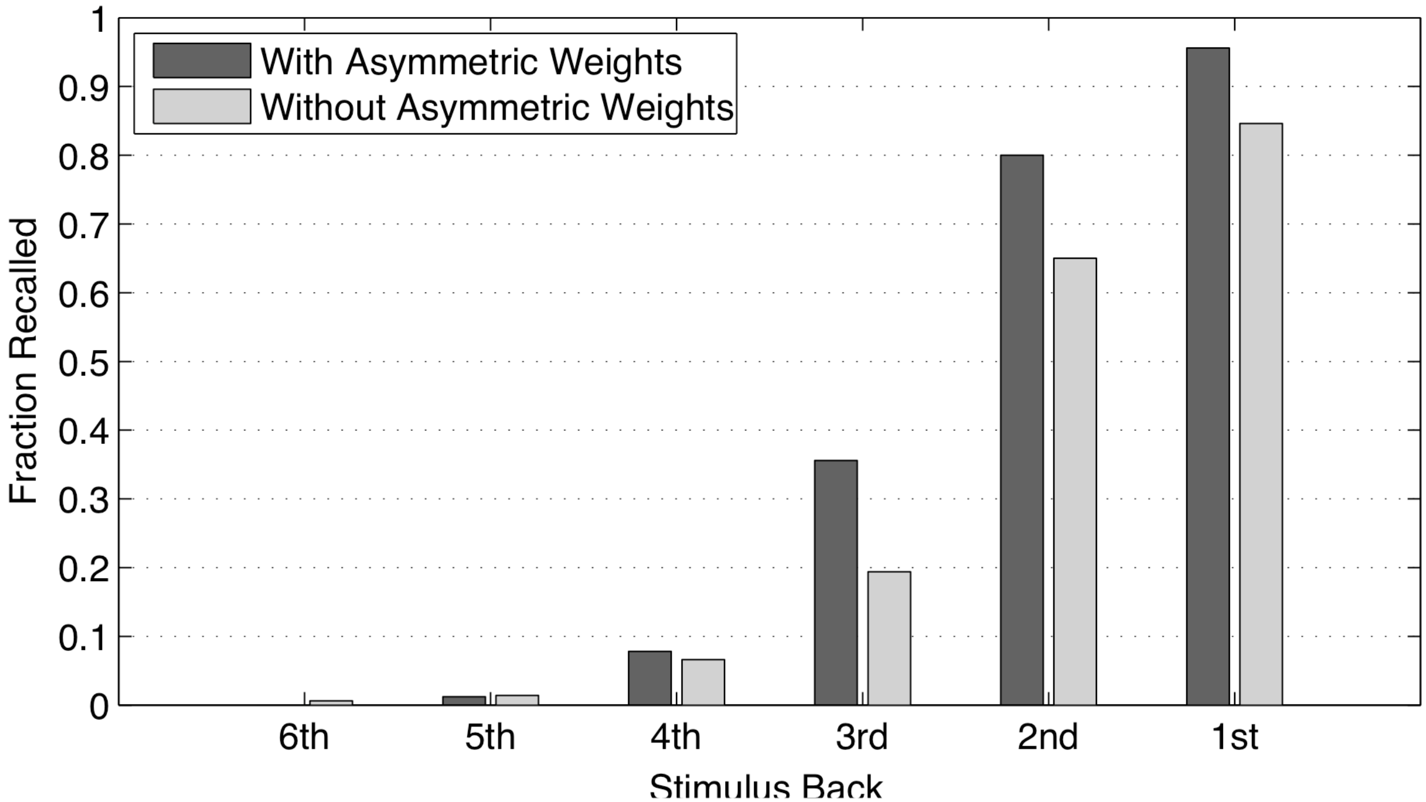
(V: asymmetric weights)

		β_2				
		0.0	0.25	0.5	0.75	1.0
	0.00	–	1.13	1.38	1.46	1.54
	0.25	1.18	1.84	2.01	2.22	2.12
β_1	0.50	1.44	1.91	1.89	2.04	2.26
	0.75	1.72	1.88	1.95	2.02	2.08
	1.00	1.76	1.90	1.93	1.93	1.85

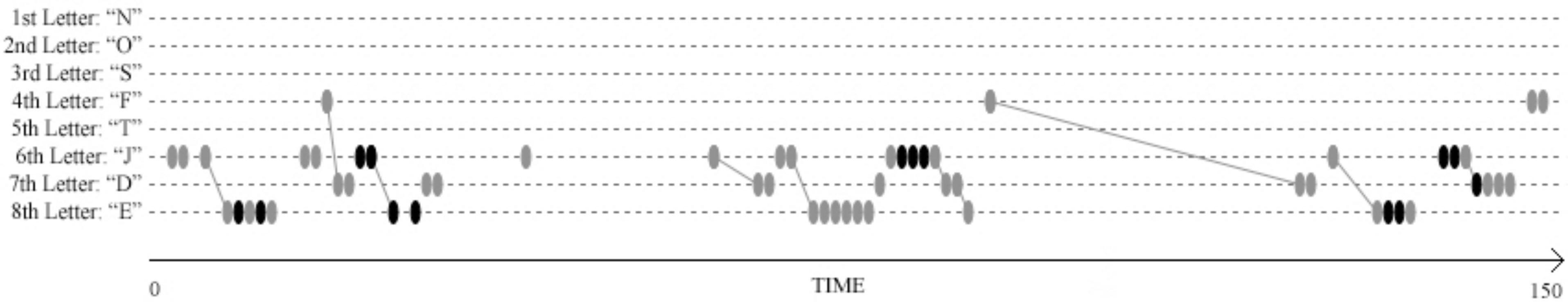
(Human subjects recalled 2.69 stimuli)

(W: symmetric weights)

Recall rates by position

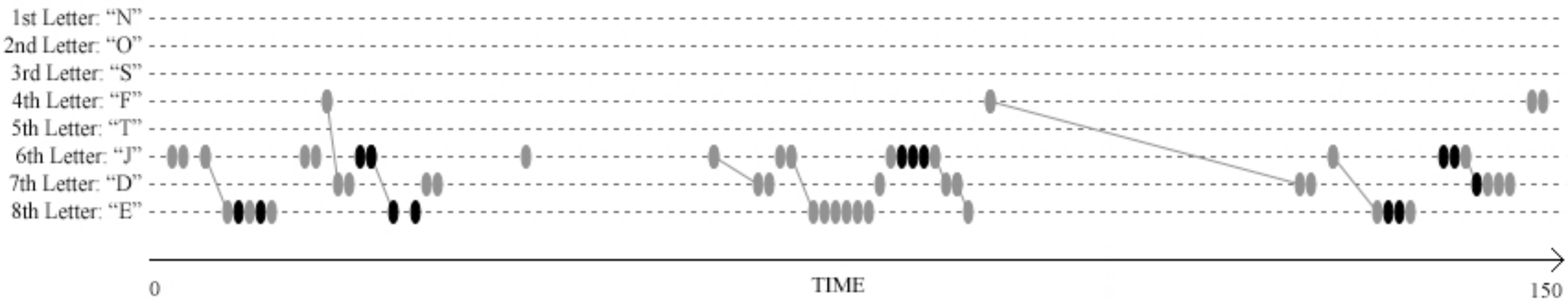


Ordering of similarity peaks

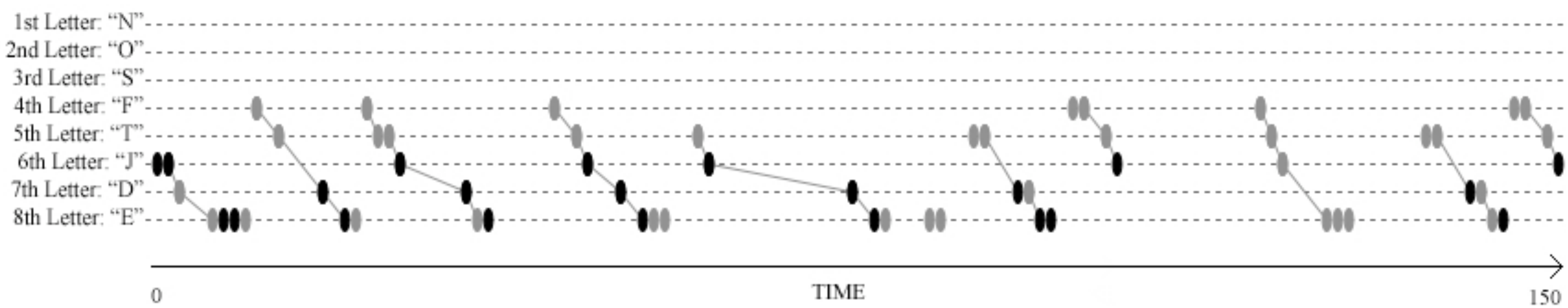


Without asynchronous weights

Ordering of similarity peaks



Without asynchronous weights



With asynchronous weights

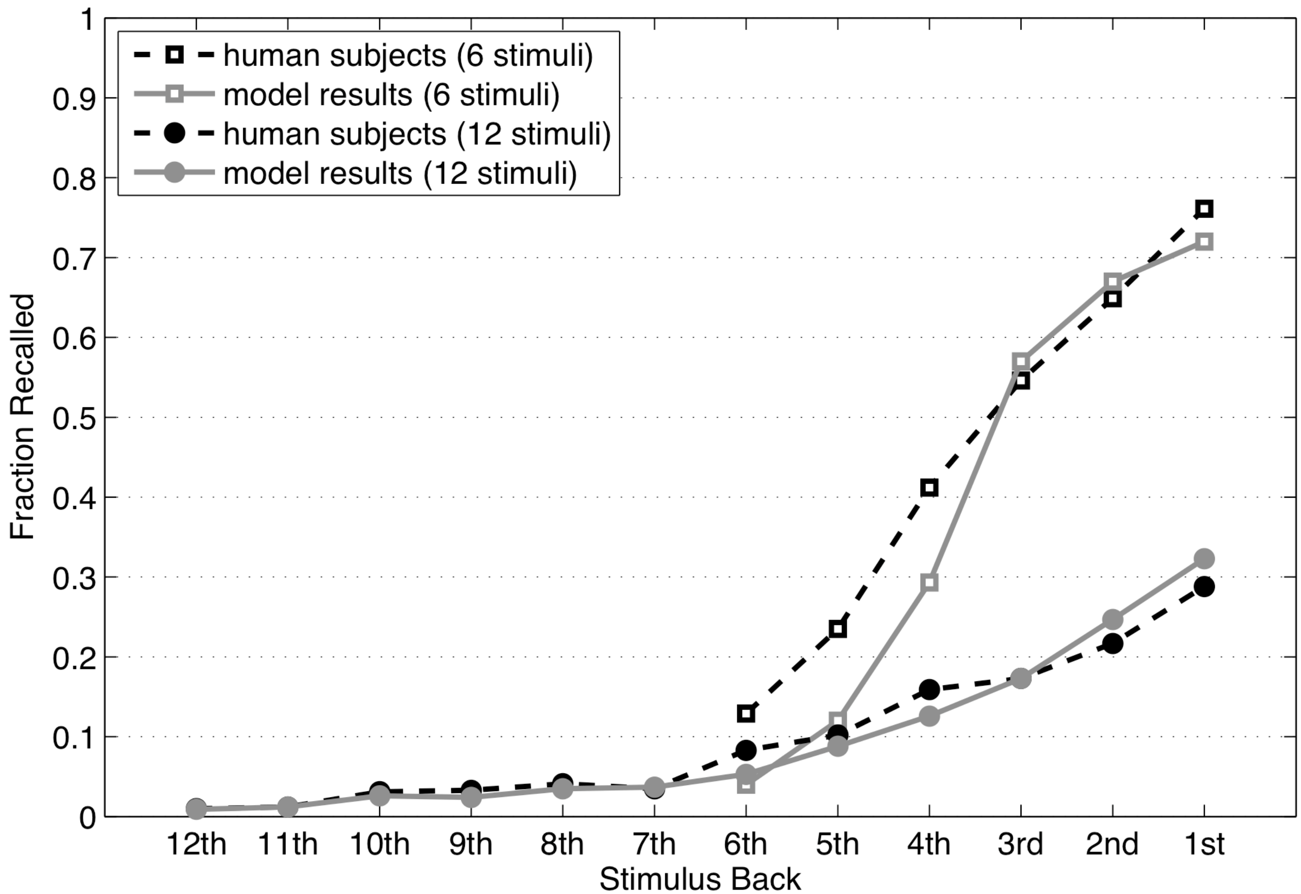
Proportion of forward peak-to-peak transitions

(asymmetric weights)

		β_2				
		0.0	0.25	0.5	0.75	1.0
	0.00	–	.81	.86	.93	.87
	0.25	.56	.71	.71	.83	.78
β_1	0.50	.50	.70	.68	.79	.85
	0.75	.56	.65	.68	.75	.78
	1.00	.53	.61	.67	.74	.71

(symmetric weights)

Comparison of model to human data



Discussion

- model now recalls stimuli in order (roughly)
  - without loss of memory capacity
  - can still match human performance
- reminiscent of chaining
  - can't be "knocked out" of sequence
- simplicity
  - no complex architecture
  - no need to store the sequence in toto
    - sequence can be reconstructed from each pair of time steps