# Recap

- Serial position curve
- Errors
- List-method directed forgetting
- Event boundaries
- Free recall brain hack

# Models of free recall

PSYC 51.09: Human Memory
Spring 2022

Jeremy Manning
jeremy@dartmouth.edu

# Putting it all together

- We've discussed many memory models so far:
    - Strength theory + variants
    - Attribute models, memory traces
    - Hopfield networks
- To explain free recall, we'll use bits and pieces of all of the above…

# What does it mean to “explain” free recall?

- If you know the specific list of words someone studied, our goal is to predict:
  - Which word they’ll remember first
  - The order they’ll recall the words in (and the timing)
  - Their overall memory performance
  - The errors they make

# We’ll consider two types of models

- **Buffer models** (“dual-store” models that have short and long term memory)

- **Retrieved context models** (“single-store” models that do not distinguish short and long term memory)

- The models have a lot in common, but they make different predictions about how time and distractions should affect memory

# We’ll consider two types of models

- We’re now at the point where the models will become too cumbersome to work through on paper

- Instead, we’ll develop intuitions for how the models work (leveraging what we’ve learned from other models we’ve worked with previously)

# The Search of Associative Memory (SAM) model

- Dual-store model proposed by Atkinson and Shiffrin (1968)
- Hugely influential and famous model– introduced the idea of separate short and long term memory systems
- Inspired in part by computer systems with separate short term and long term storage (RAM vs. hard disk)
- Also inspired by clinical finding that amnesic patients appear to have normal short term memory but cannot form new long-term memories
- Around the same time, Schwartz and Kandel identified two distinct molecular mechanisms supporting memory (Kendel got the Nobel prize for some of this work in 2000). "Short-term memory" was supported by recruiting new proteins to the synapse. "Long-term memory" was supported by protein synthesis.

# Dual-store models

- **Short-term storage:** a working memory buffer (easy to access and modify, but limited in capacity)
- **Long-term storage:** memories are linked to each other (and to their contexts)

# Aside: fast learning = fast forgetting

- Why might we benefit from these two types of storage?
- Think about Hopfield networks: what happens when you change weights to store a memory?
- If we change the weights a lot, we learn quickly but disrupt (forget) already-stored memories.  If we change weights a little we learn slowly but forget less.
- One way to accomplish both fast learning and slow forgetting would be to have two separate systems that were tuned differently

# The Search of Associative Memory (SAM) model

- There are many parts to the model– I’ll describe the key elements first, and then we’ll step through an example

# The Search of Associative Memory (SAM) model

- SAM assumes we form two types of associations:
  - **Episodic**: associations between memories and their contexts. Contexts change over time during the experiment. Association between an item and context: $S_e(i, context)$. Association between two items: $S_e(i, j)$.
  - **Semantic**: pre-existing knowledge about the meanings of words (and how they relate to each other) that doesn’t change during the experiment. Association between two items: $S_s(i, j)$.

# The Search of Associative Memory (SAM) model

- First we need to describe how the long and short term memory stores will work…

# Long term memory

- Long-term episodic memory storage (LTS = long term storage)
- Idea: stores how items and contexts are associated
- How much are *i* and *j* associated? $S_e(i, j)$
- How much is *i* associated with context? $S_e(i, context)$

# Short term memory

- AKA: rehearsal buffer, working memory, short term storage (STS)

- Intuition: we want to describe what happens when we hold an item in mind.  What happens is that we form associations between that item and other things in mind, and between that item and a context representation.

# Short term memory

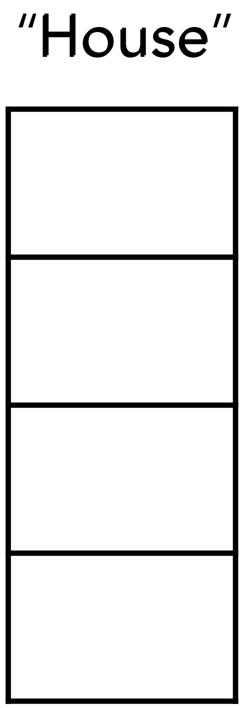
- A set of 4(ish) items that are considered *active*
- When you study an item, it becomes active. If the buffer is full, a probabilistically chosen active item gets deactivated.
- Active items (in STS) are strengthened in LTS:
    - When an item is active, it's associate with context, $S_e(i, context)$, is increased
    - When two items are co-active, $S_e(i, j)$ is increased

# The Search of Associative Memory (SAM) model

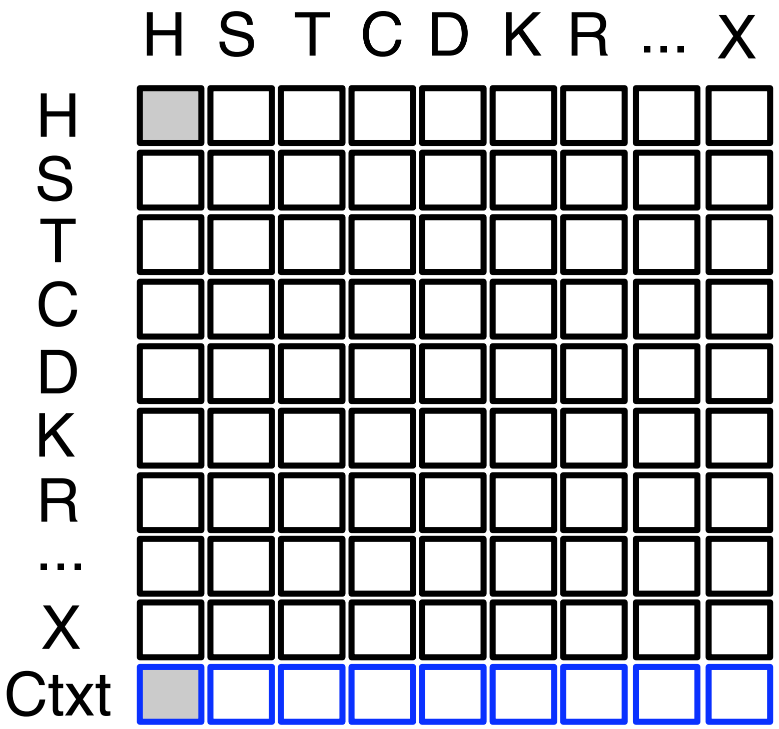
- Next: example showing how studied items enter (and exit) the short term memory buffer, how long term memory is affected, and how these processes are modulated by semantic memory

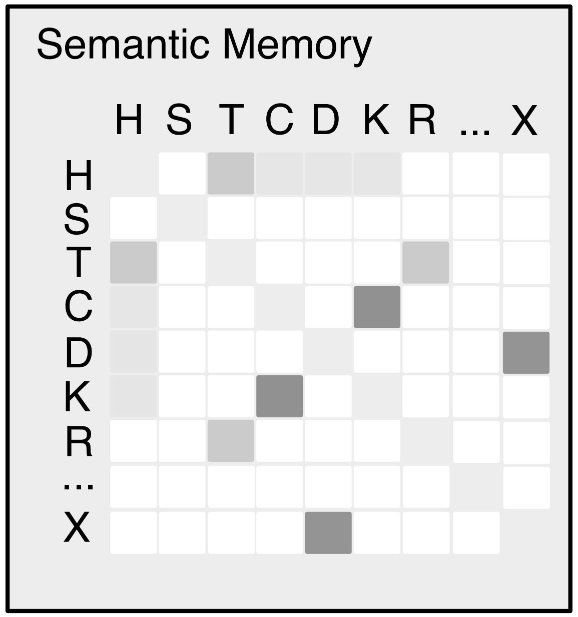
# Encoding in SAM

## STS



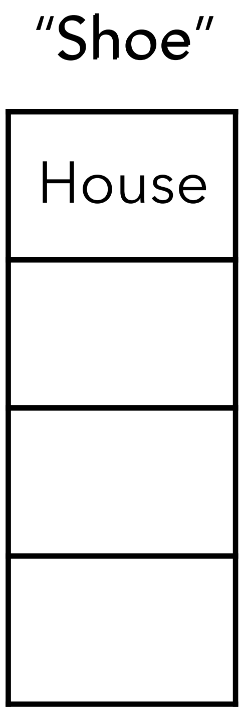
## Episodic LTS



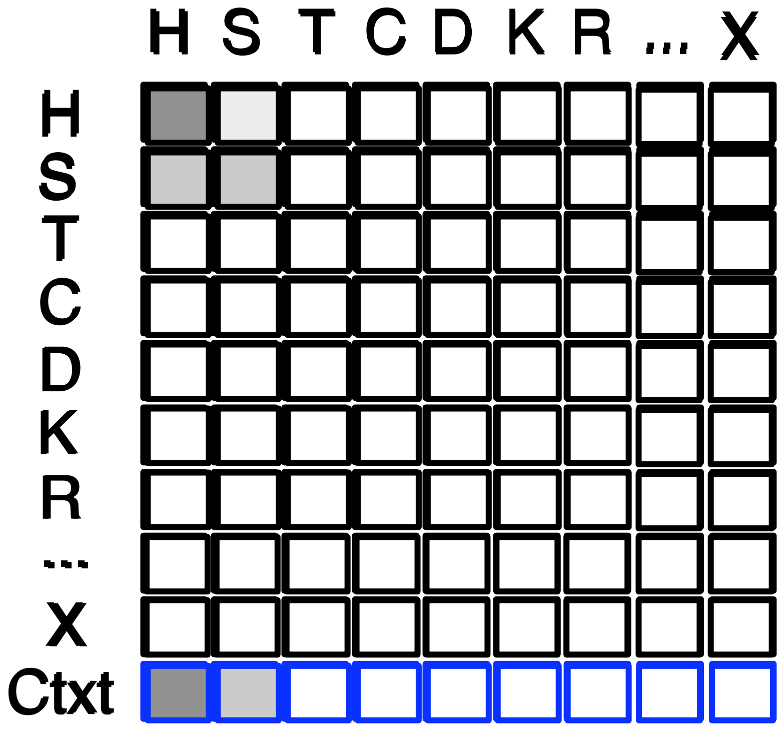


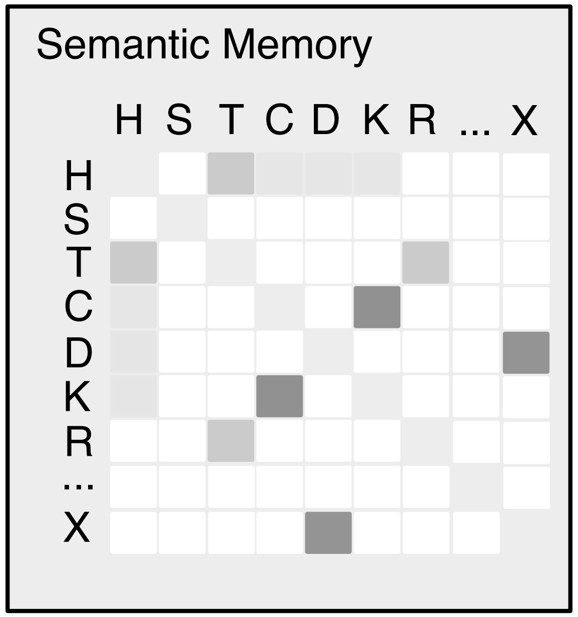
# Encoding in SAM

## STS



## Episodic LTS

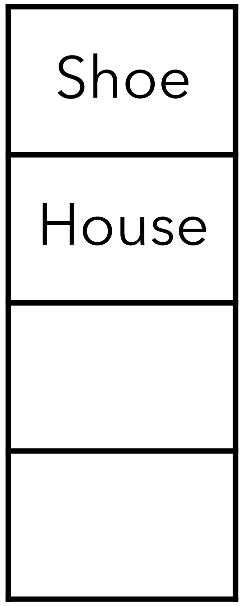




# Encoding in SAM

STS

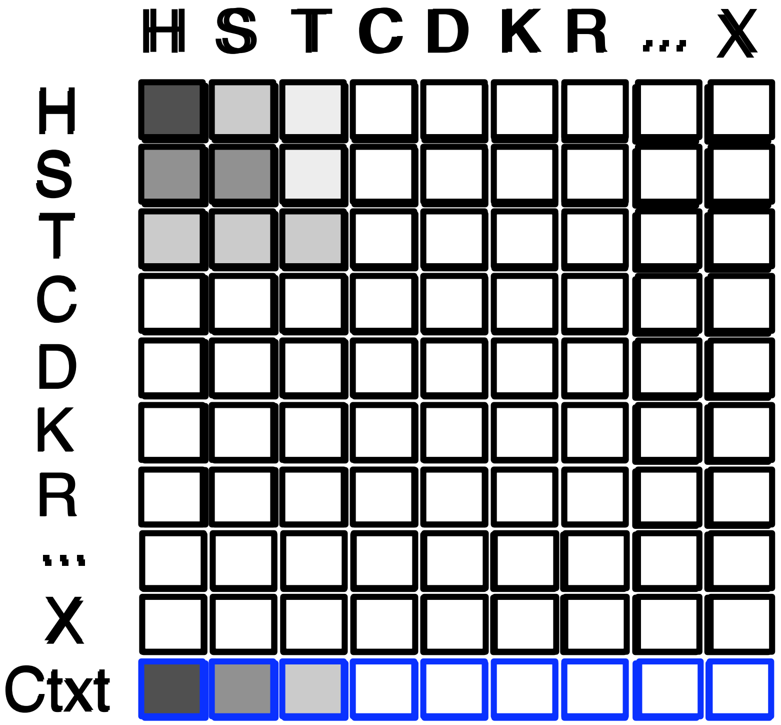
"Tree"

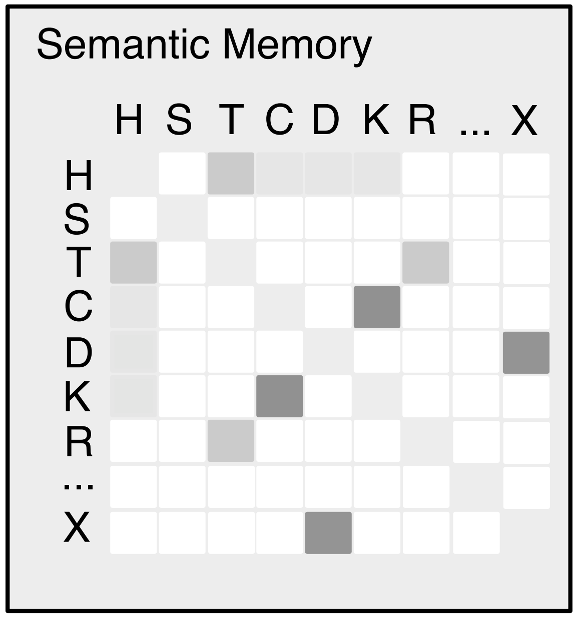


Shoe

House

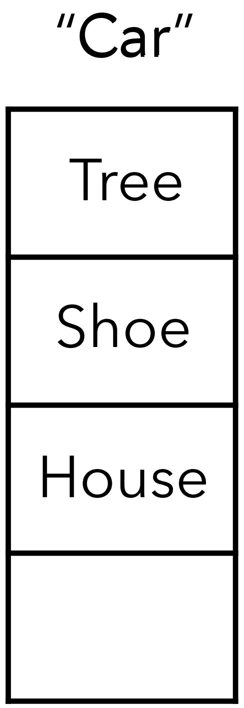
Episodic LTS





# Encoding in SAM

## STS



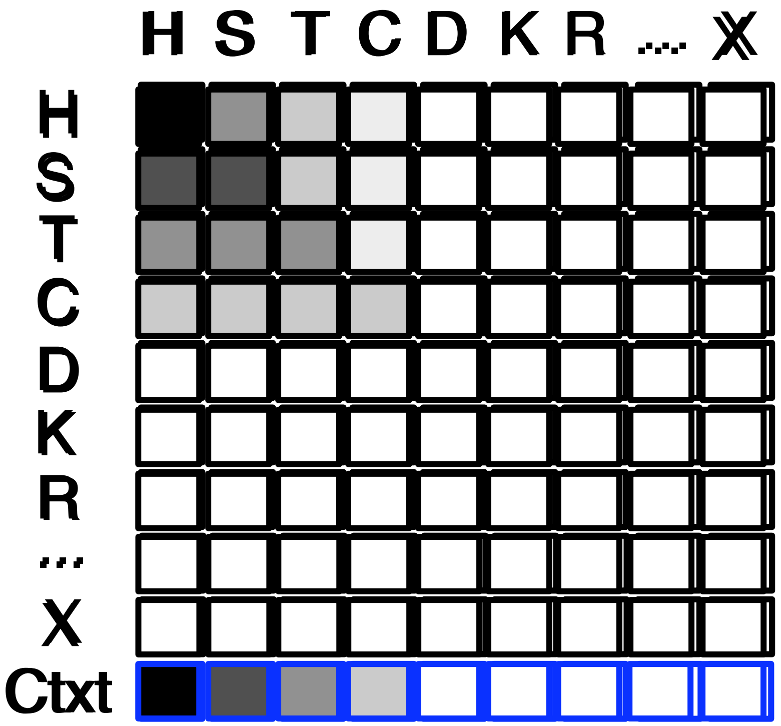
"Car"

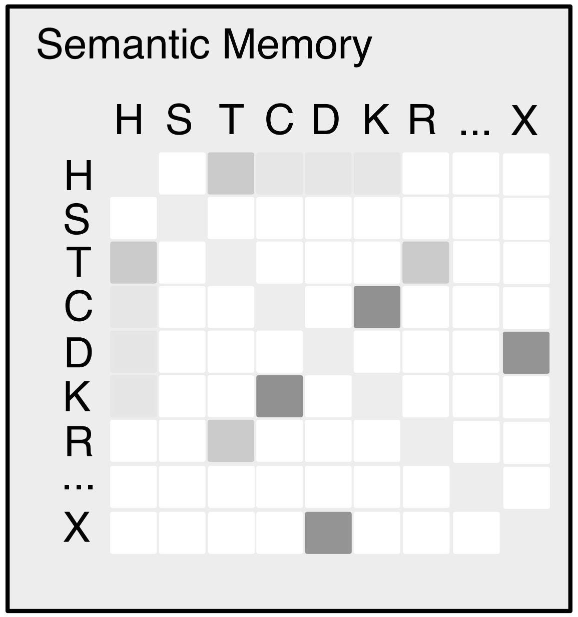
Tree

Shoe

House

## Episodic LTS





# Encoding in SAM

## STS

"Dog"

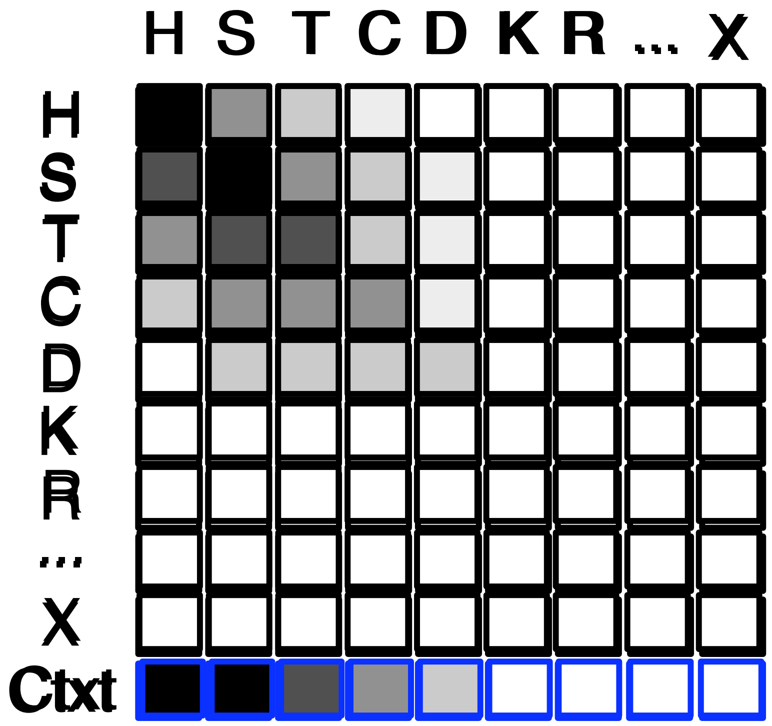
Car

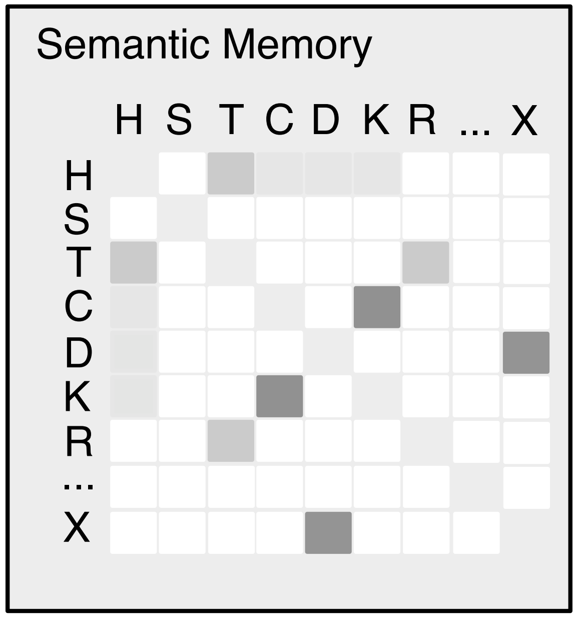
Tree

Shoe

House

## Episodic LTS





X

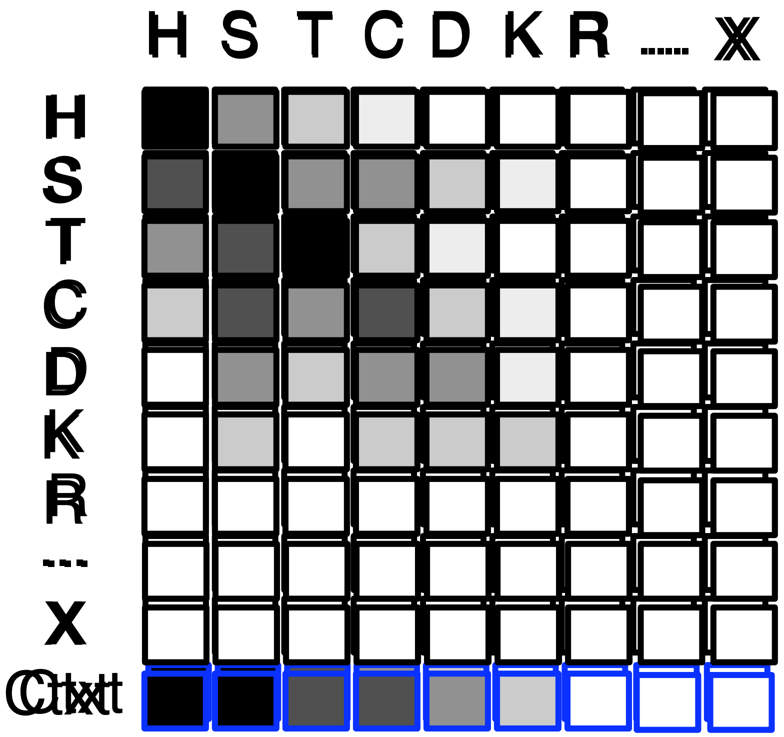
# Encoding in SAM

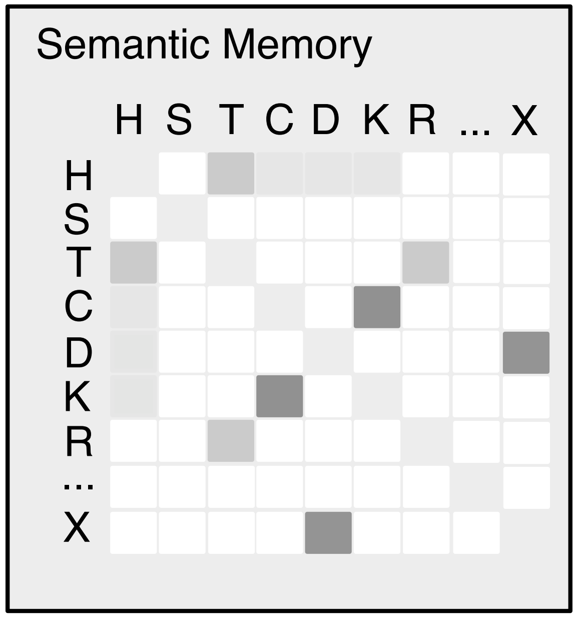
## STS

"Key"

- Dog
- Car
- Tree
- Shoe

## Episodic LTS





X

# Encoding in SAM

STS

"Rose"

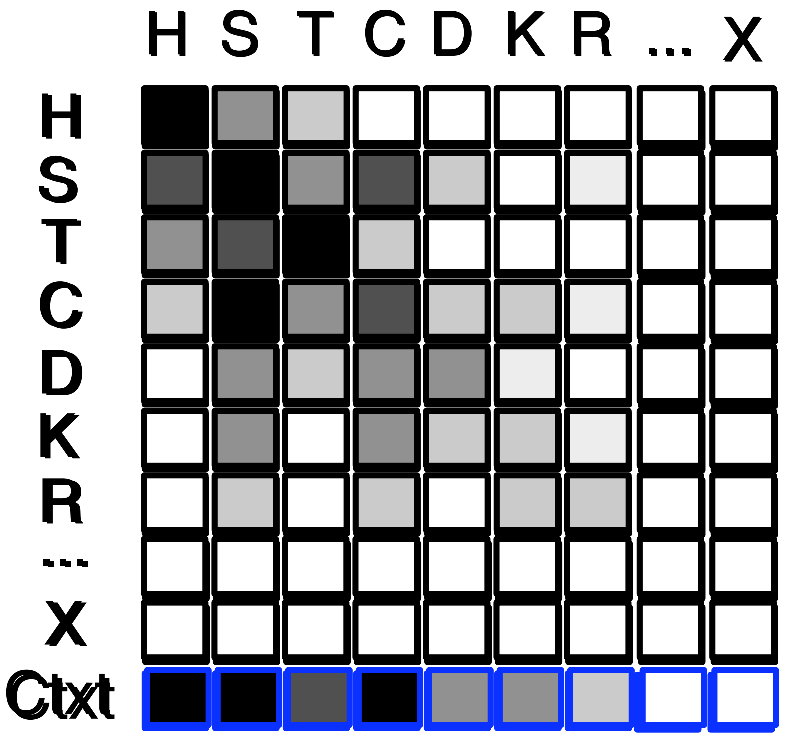
Key

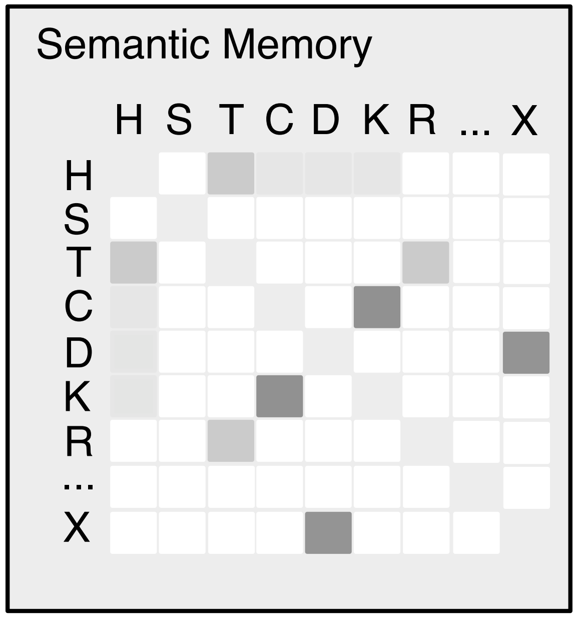
Dog

Car

Shoe

Episodic LTS





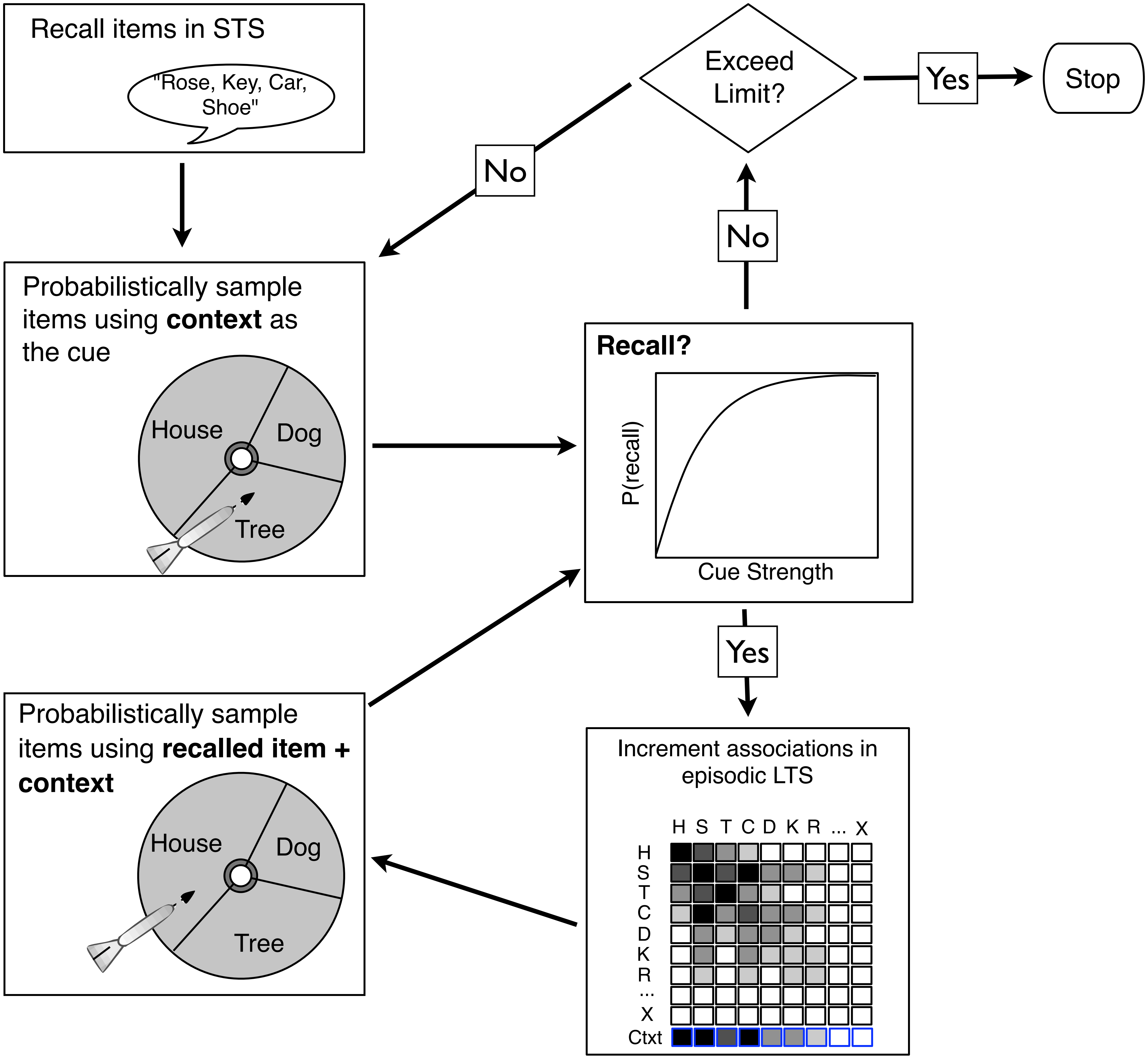
X

End of list

# The Search of Associative Memory (SAM) model

- Now we’ve finished studying the list, and that’s affected the short term and long term stores

- How do we read the information back out?


Recall items in STS
"Rose, Key, Car, Shoe"
Exceed Limit?
Yes
Stop
No
No
Probabilistically sample items using **context** as the cue
House
Dog
Tree
Recall?
P(recall)
Cue Strength
Yes
Probabilistically sample items using **recalled item + context**
House
Dog
Tree
Increment associations in episodic LTS
H S T C D K R ... X
H
S
T
C
D
K
R
...
X
Ctxt

# Fitting SAM to free recall data

- The parameters of SAM control things like
  - How much association strengths change (in LTS and context) for items in the STS
  - Relative weightings of context vs. semantics vs. LTS in retrieval
  - For a given set of parameters, you can have the model study and recall the same lists as a human participant

# Fitting SAM to free recall data

- Idea: adjust model parameters so that the model produces data that look like what people actually did in the experiment

- Note: this isn't guaranteed to work. We may find that *no* combination of parameters will produce data that look like what people actually did

- Finding a good combination of parameters means the model "predicts" (read: "retroactively explains") people's behaviors

# Model fitting procedure

- Initialize model parameters (e.g. make reasonable or random guesses)

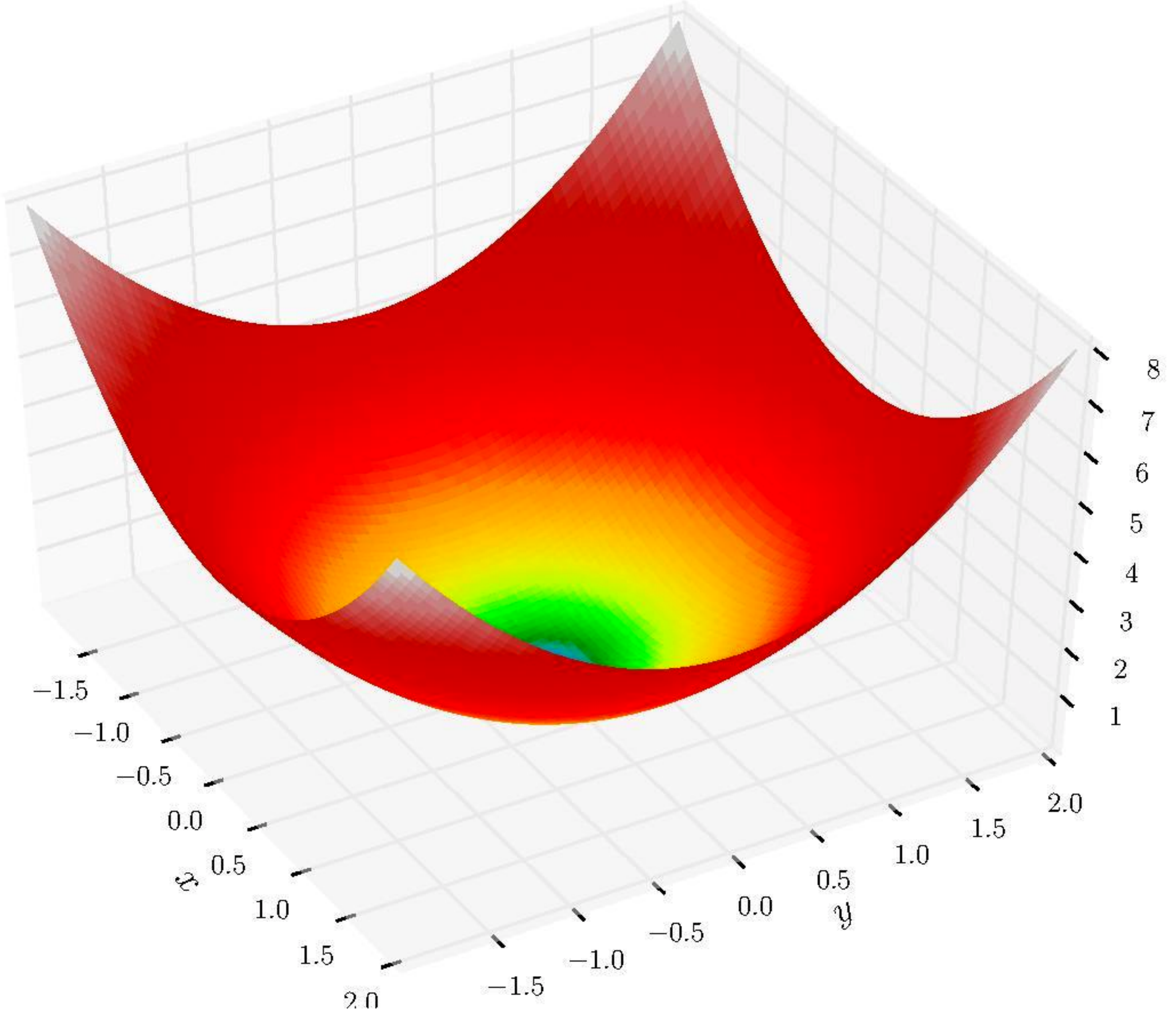
- Run a simulation to determine what the model predicts the data should look like (e.g. generate performance curves and summary stats)

- Ask: how far off are the predictions? Error = $(\text{model} - \text{data})^2$

  - Think: feature vectors!

- Repeat this process (tweaking parameters each time) until the error is minimized

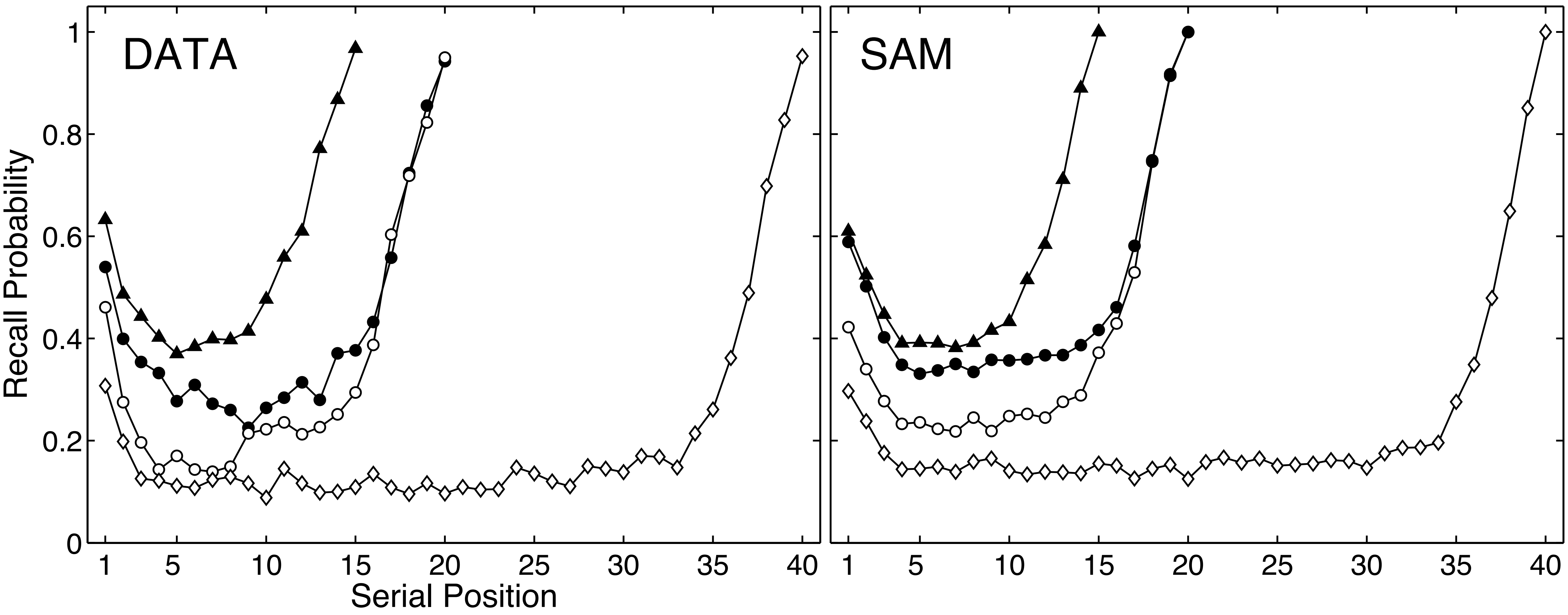
# Model fitting procedure



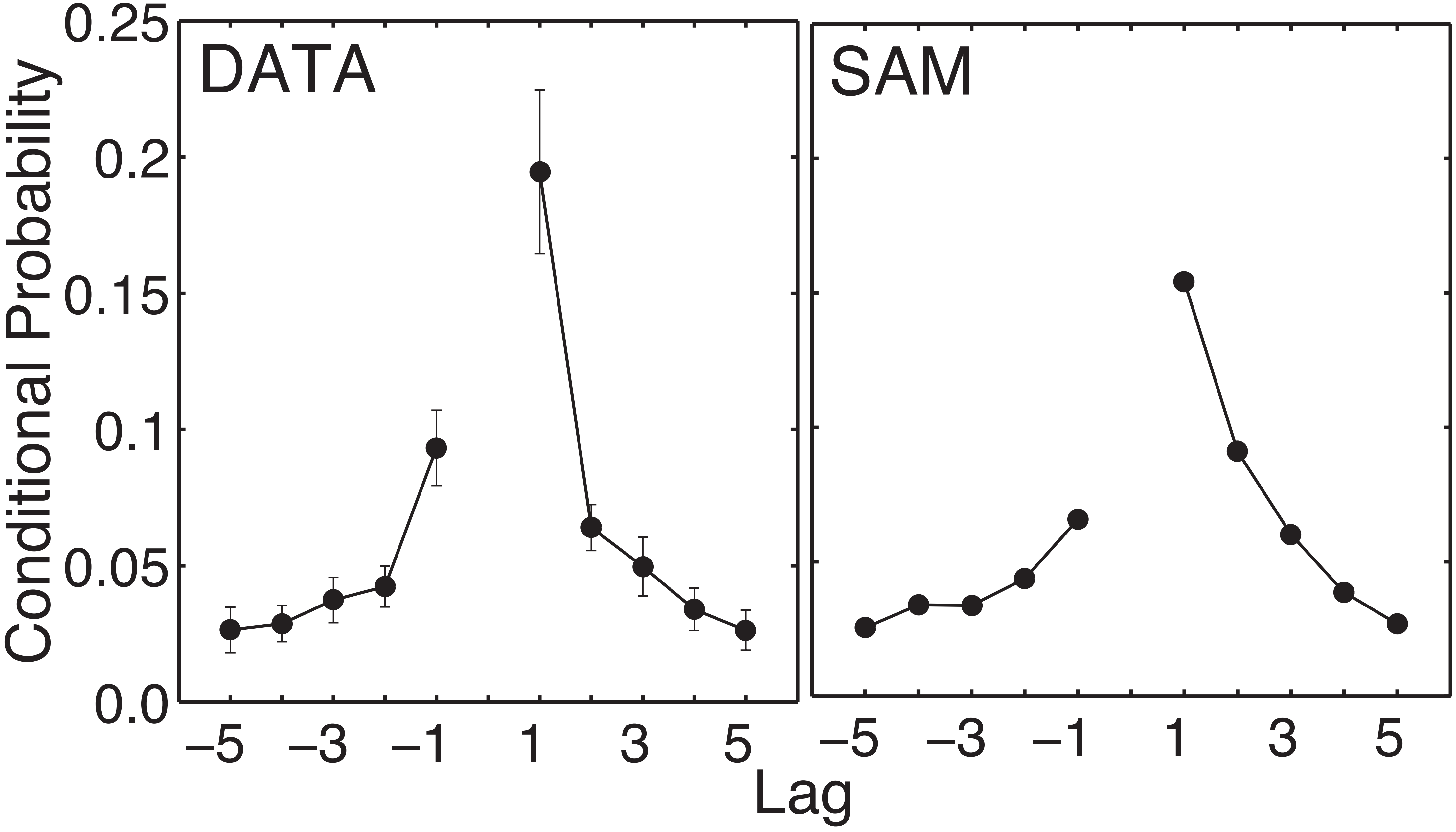
# What sorts of things will we want to predict?

- Anything that characterizes people's free recall behaviors
- The more we're able to predict (explain) with the model, the better the model is

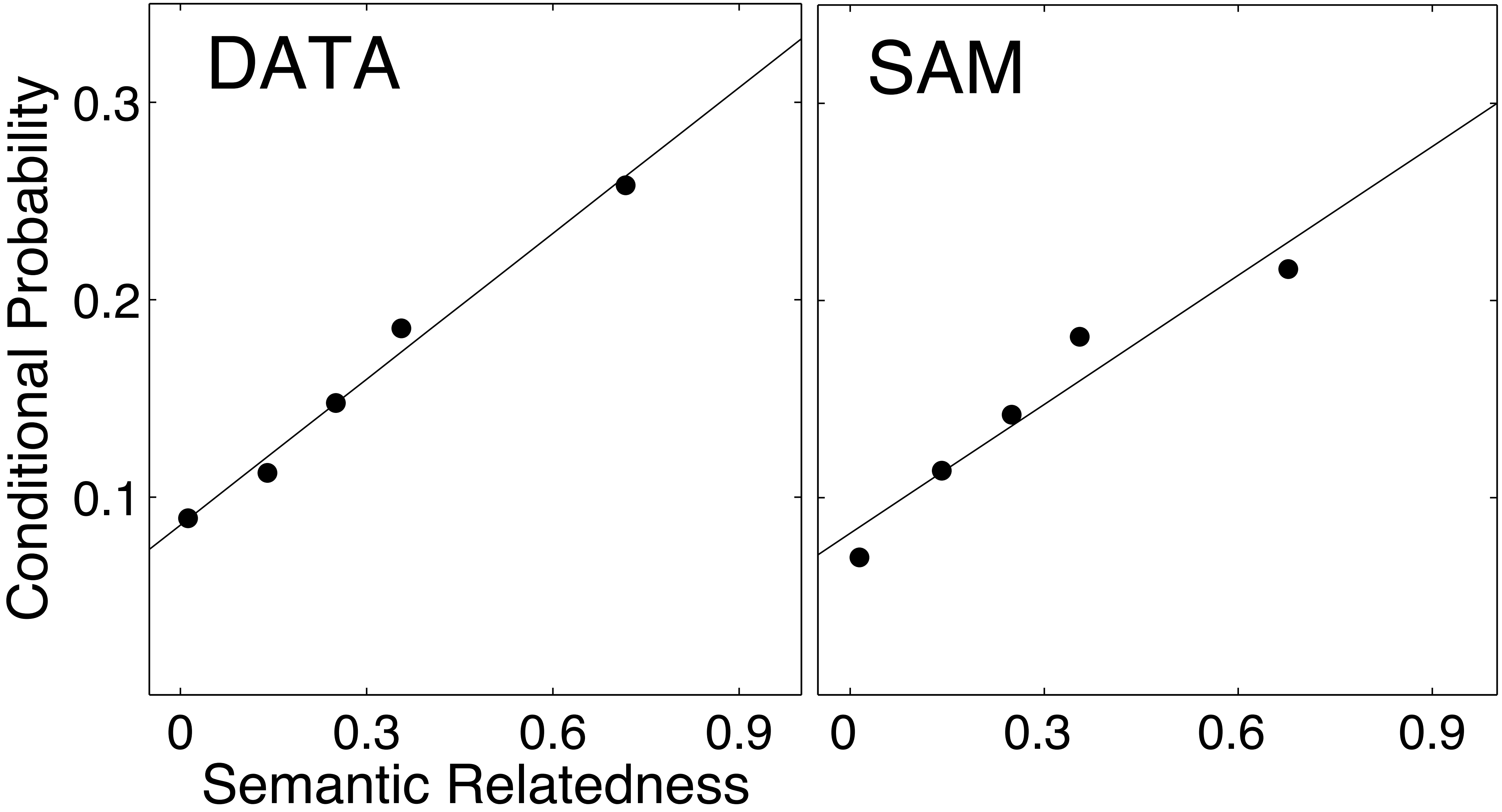
# Serial-position curves for lists of different lengths



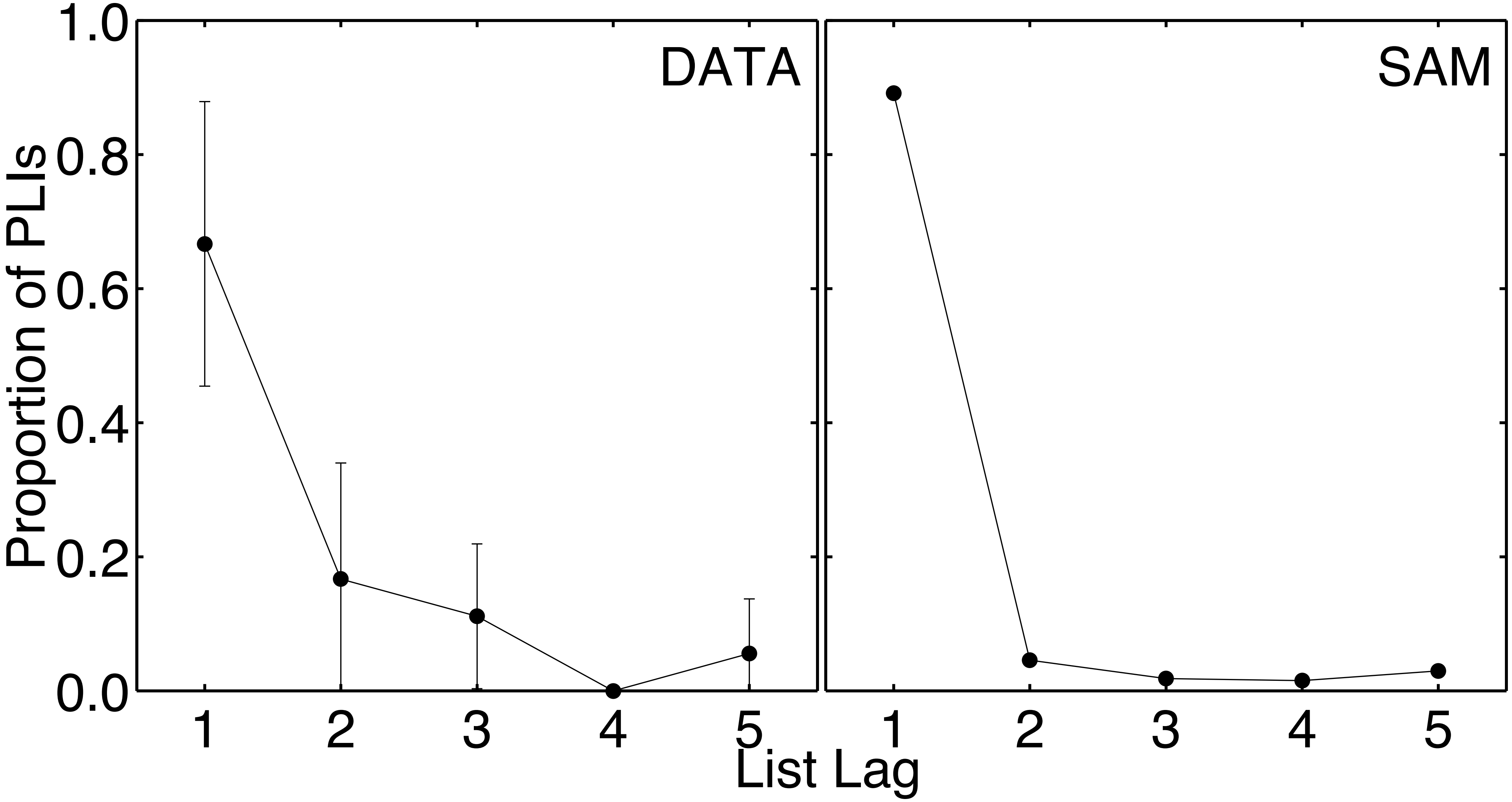
# Temporal clustering



# Semantic clustering



# Intrusions



| | Correct | PLI | XLI |
|---|---|---|---|
| Howard & Kahana (1999) | 5.0 (0.50) | 0.5 (0.2) | 0.2 (0.1) |
| SAM | 5.0 (0.05) | 0.47 (0.02) | 0.14 (0.01) |

# Things SAM can predict

- Primacy and recency effects: p(first recall) and overall recall, for different list lengths

- Transition effects: temporal and semantic clustering

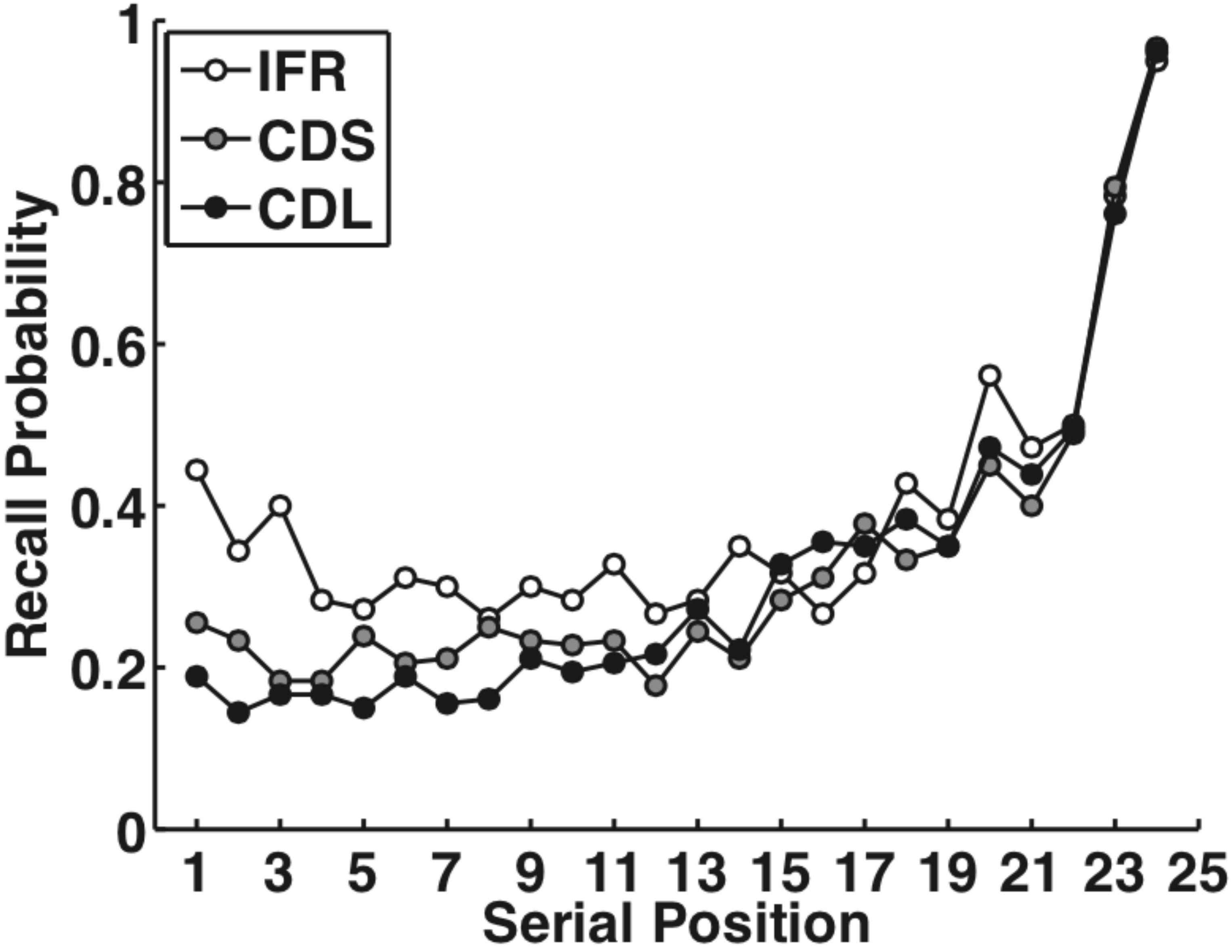
- Errors: prior list intrusions and extra list intrusions

# Things SAM can predict

- Recency effect goes away with delayed free recall: buffer is filled up by the distractor, so the contents of STS isn't output first.

- But…what happens with continual distractor free recall?

# Lists of categorized items, 0, 2.5, or 7.5 sec of continual distraction


IFR
CDS
CDL
Recall Probability
Serial Position