Patience is a virtue: Learning when to encode/recall episodic memories

Qihong Lu[1], Zi Ying Fan[2], Uri Hasson[1], Kenneth A. Norman[1]

1. Department of Psychology and Princeton Neuroscience Institute, Princeton University; 2. Computer Science Department, Princeton University

Main point

Q: How should encoding and retrieval processes be parameterized to support event prediction?

We found that a certain amount of waiting, for both encoding and retrieval, reduces the probability of false recall, which boosts the extent to which episodic memory benefits event understanding.

An episodic neural network

Cortex: a recurrent neural network (LSTM)

- predicts the next event, $\hat{s}_{t+1}$
- sends ϑ_t to hippocampus (EM) to configure the parameters of the LCA (fig 1 B)

Hippocampus (EM)

- **recall**: given the current cortical pattern c_t and ϑ_t, return a memory, m_t, to the cortex
  - the recall process is governed by a LCA (fig 1 B).
- **encode**: save the cortical pattern, c_t

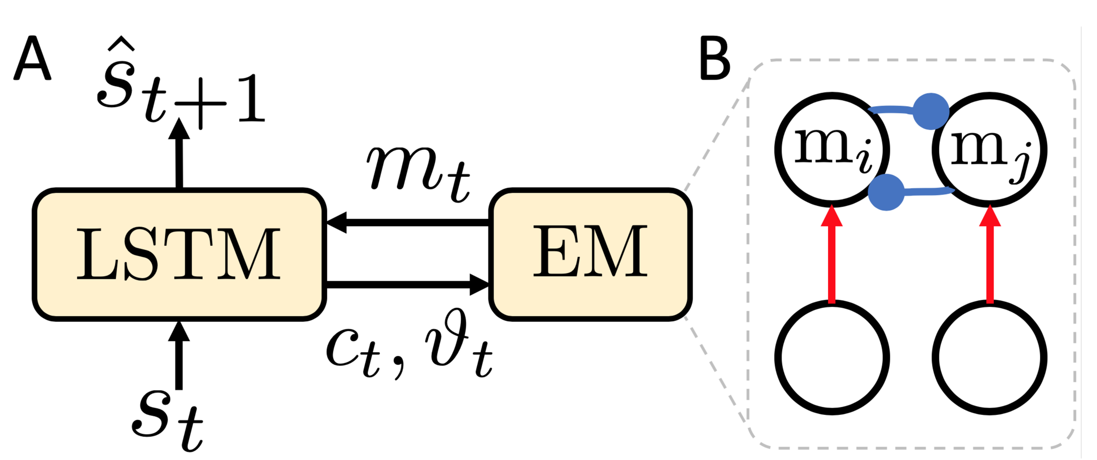

Figure 1: A) A recurrent neural network with episodic memory (EM); B) A **leaky, competing accumulator (LCA)**. For a memory (e.g. m_i) to be recalled, it needs to be highly activated to counteract the leak term, and outcompete other memories (e.g. m_j). **The level of leak and competition are controlled by cortex via ϑ_t**; Encoding new memories = adding new nodes to LCA.

References & Acknowledgement

[1] Baldassano, C., et al. (2017) Neuron.
[2] Ben-Yakov A., & Dudai, Y. (2011). J Neuro.
[3] Ben-Yakov A., & Henson, N. (2018). J Neuro.

Acknowledgement: This work was supported by a Multi-University Research Initiative grant to KAN and UH (ONR/DoD N00014-17-1-2961). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research or the U.S. Department of Defense.
Download the poster: `https://tinyurl.com/cems19-q`

A recall/no-recall task

- **Encoding phase**: Generate k event sequences from the event graph (fig 2 A).
  - To generate an sequence, we randomly sample a "situation", which defines a path on the event graph.
  - Each transition on the graph is controlled by a particular feature of the situation. Thus, knowing the feature values makes it possible to predict what will happen next.
- **Test phase**: Flush the cortical activity (the LSTM hidden state). With $p = .5$, present a previously seen event sequence with a different order (a **recall trial**, e.g. fig 2 B); otherwise, present a new sequence (**a no-recall trial**).

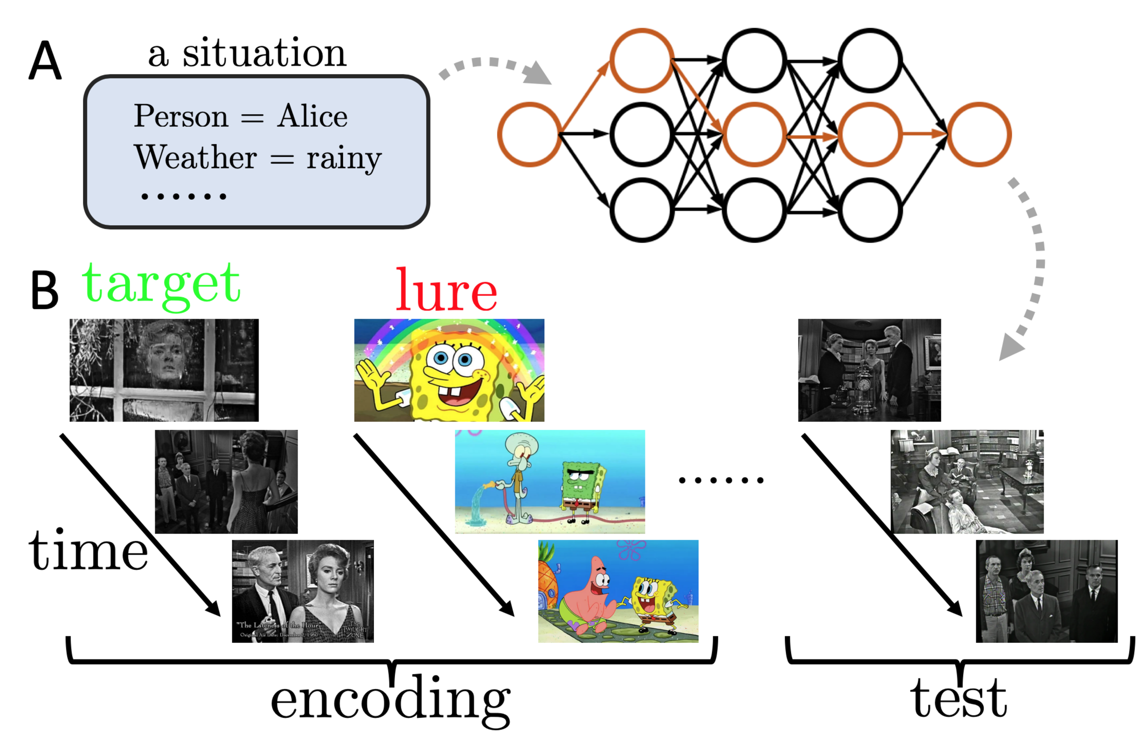

Figure 2: A) Sample an event sequence (in brown) from the event generative model; B) A recall trial.

After training, the model learned to use episodic memories to support event prediction (fig 3), suggesting that feature values (e.g. weather = rainy) of the situation are encoded in the episodic memories.

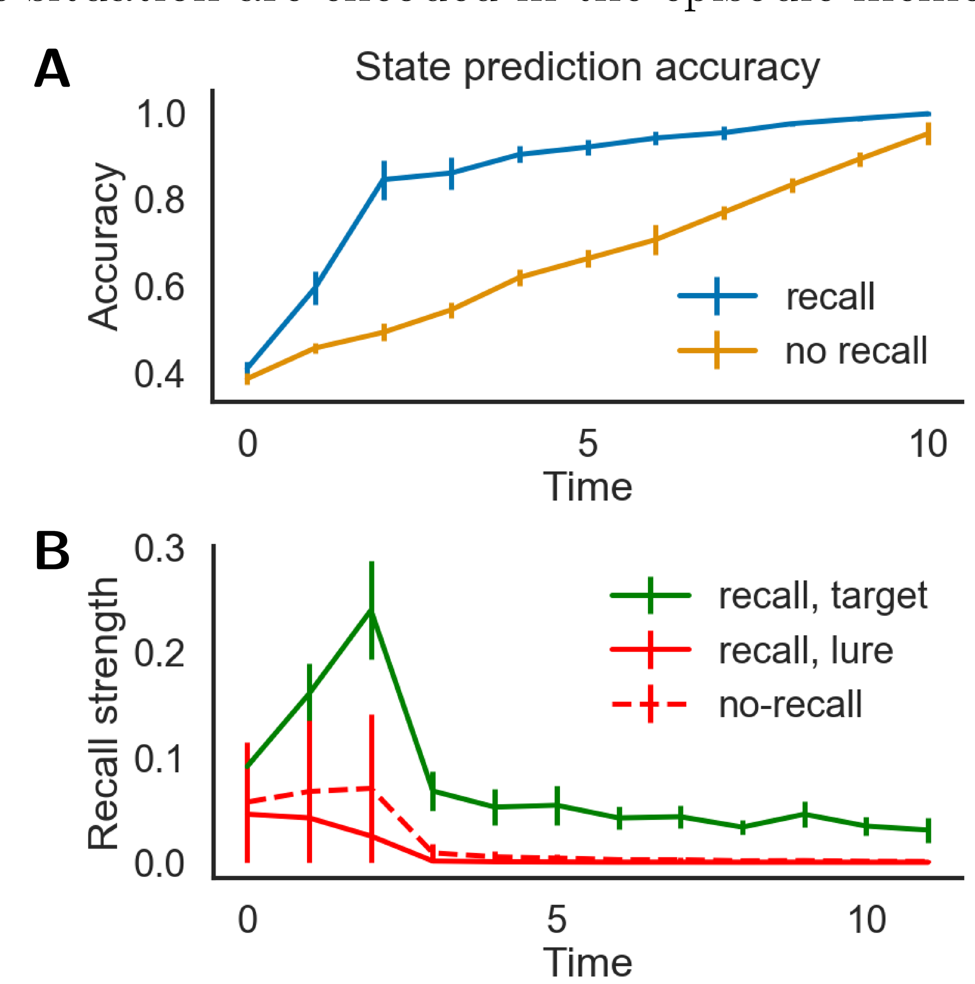

Figure 3: A) Prediction accuracy is higher for recall trials; B) Target memories are more activated than lures.

Waiting to recall

Typically, the model needs to choose between...

- **recalling early**, with more risk of recalling lures
- **waiting to recall**, but the prediction benefit of recall diminishes over time.

When the need to predict is delayed (fig 4), the optimal strategy is to wait to recall until the need to predict arises; recalling earlier carries only risk (of false recall) but no reward. We found that the model learned to wait to recall (fig 5) to avoid recalling lures.

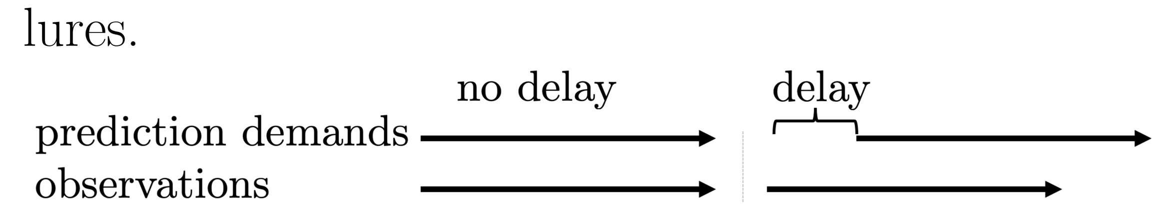

Figure 4: For the delayed condition, initially, the model receives observations of the current situation without being queried about what happens next.

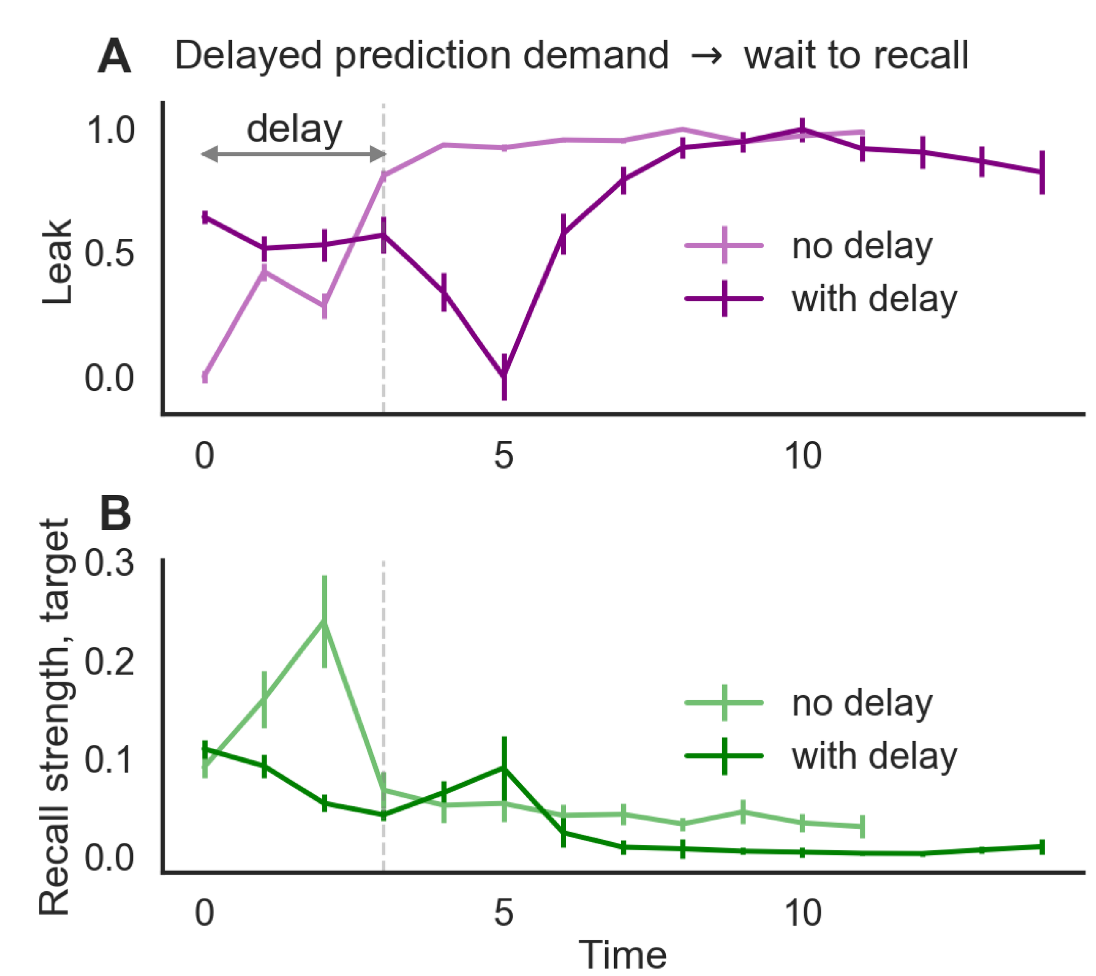

Figure 5: The model learned to wait to recall when prediction demand is delayed. A) The leak value (a LCA parameter controlled by the LSTM) over time. Leak governs how likely memories are to be recalled. Smaller leak values indicate stronger recall; B) The recall strength of the target memories.

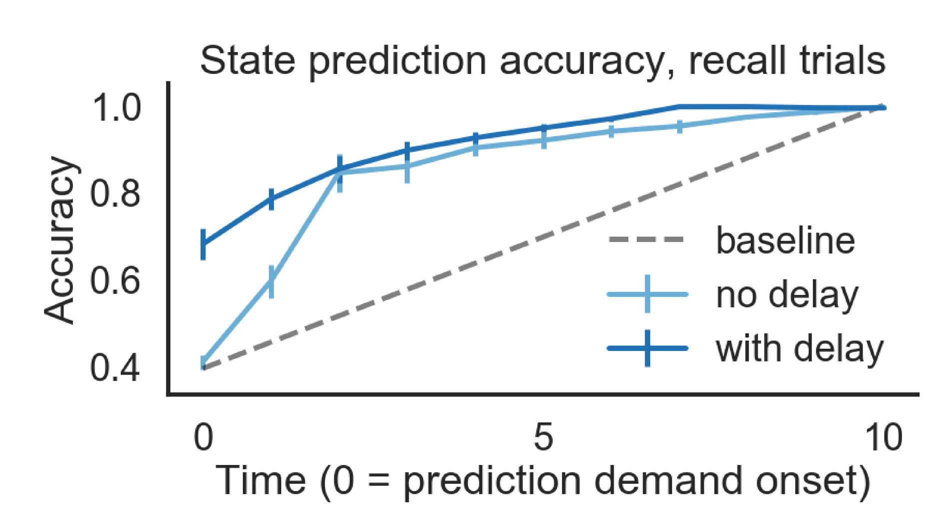

Figure 6: Prediction accuracy, no delay vs. delay. Time courses are aligned to prediction demand onset.

Encode at event boundaries

We considered three encoding regimes (fig 7 A.):

a. **non-overlapping small chunks**: store non-overlapping memories at the sub-event level
b. **cumulative**: memories have temporally-nested structure within events
c. **encode at event boundaries**: wait until the end of event and store a single memory

We found models that encode at event boundaries performed the best at subsequent recall (fig 8).

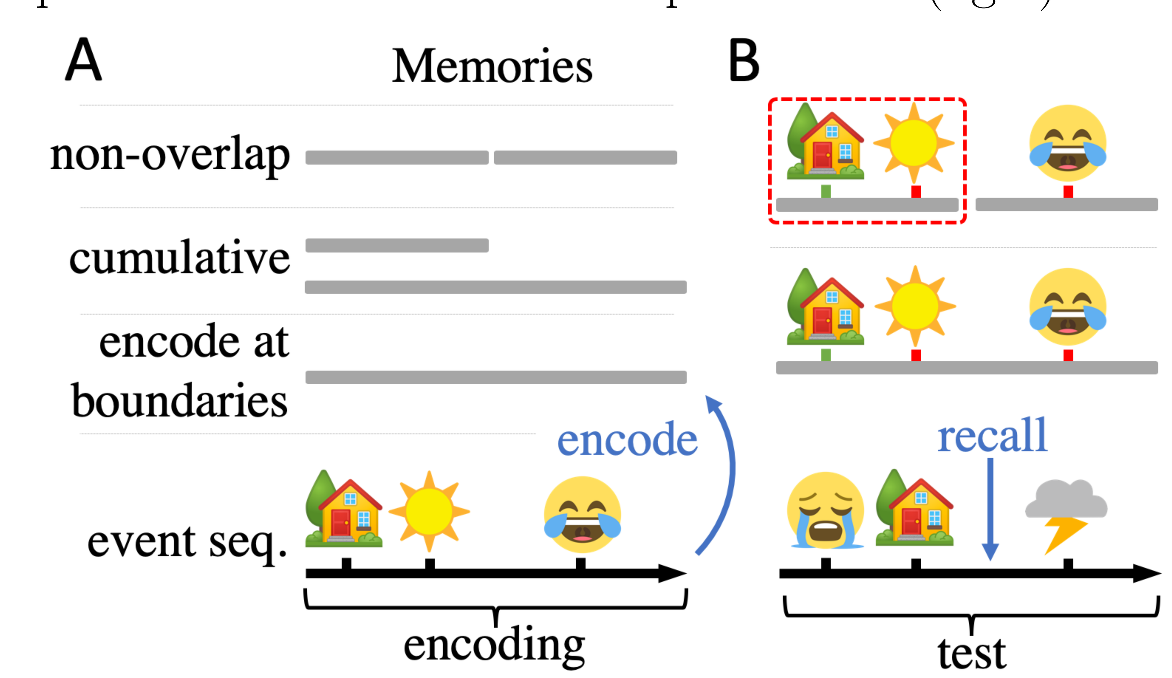

Figure 7: A) The resulting memory chunks under the three encoding regimes; B) "Small-chunk encoding" might cause subsequent false recall (boxed in red). When the agent has partial knowledge about the current situation (e.g. location = the house, mood = sad), lures are easier to reject if all information is connected.

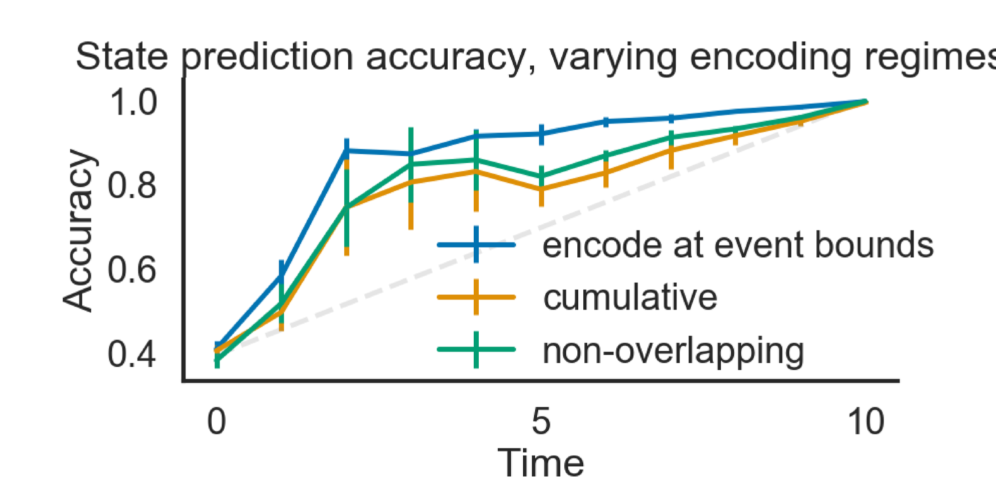

Figure 8: Models that encode at event boundaries had better event prediction accuracy.

Summary

To help with event prediction, an agent should wait to recall until prediction demand arises, and encode at event boundaries. These results provide a normative account of the observation that the neural signature of encoding is temporally sparse and time-locked with event boundaries [1][2][3].