Learning to use episodic memory for event prediction

Qihong Lu, Uri Hasson, Kenneth A. Norman

Department of Psychology and Princeton Neuroscience Institute, Princeton University

Author Note

Qihong Lu https://orcid.org/0000-0002-0730-5240

Correspondence concerning this article should be addressed to Qihong Lu, Department of Psychology, Princeton University, NJ, 08540. E-mail: qlu@princeton.edu

Abstract

When should episodic memories be stored and retrieved to support event understanding? Traditional list-learning memory experiments make it obvious when to store and retrieve memories, but it is less obvious when to do this in naturalistic settings. To address this question, we trained a memory-augmented neural network to predict upcoming events, in an environment where situations (sets of parameters governing transitions between events) sometimes reoccurred. The model was allowed to learn a policy for when to consult episodic memory, and we explored how this learned policy varied as a function of the task environment. We found that the learned retrieval policy is shaped by internal uncertainty about upcoming events, the level of penalty associated with incorrect predictions, the confusability of stored memories, the presence of a "familiarity signal" indicating the availability of relevant memories, and the presence of statistical regularities (prototypical events). With regard to encoding policy, we found that selectively storing episodic memories at the end of an event (but not mid-event) leads to better subsequent event prediction performance and less incorrect recall. Additionally, we found that the model can integrate information over long timescales even without the hippocampus; it can link information over many event segments via episodic memory; and it shows classic schema-consistent memory effects when the upcoming time point has a prototypical event. Overall, these modeling results provide a normative explanation of several existing behavioral and neuroimaging findings regarding the use of episodic memory in naturalistic settings, and lead to a wide range of testable predictions.

Keywords: event cognition; episodic memory; memory-augmented neural networks

Learning to use episodic memory for event prediction

Contents

Author Note **1**

Abstract **2**

Learning to use episodic memory for event prediction **3**

List of Figures **5**

Modeling targets **8**

When to recall 8

When to encode 11

A model of cortical-hippocampal interaction **12**

The cortical network 12

The hippocampus 14

Episodic encoding 14

Episodic recall 15

Episodic recall – detail 16

Objective function and learning algorithm 17

Model execution 18

Task formulation **20**

An event sequence is a sample from an event schema conditioned on a situation 20

Task conditions 22

Model training 25

Stimulus representation 26

Analysis methods **28**

Multivariate pattern analysis 28

Functionally aligning multiple neural networks 30

Brief summaries of simulations **30**

Simulation 1: basic properties of the model **35**

1.0 Decoding the internal states of the network 35

1.1 Using episodic memory to guide event prediction 36

1.2 The learned policy adapts to the average penalty associated with misprediction during training 41

1.3 Interactions between penalty and uncertainty 42

Empirical prediction 1 – how episodic retrieval is modulated by uncertainty 44

Simulation 2: The learned policy is sensitive to the current penalty level during test **44**

Simulation 3: Recall is sensitive to the interaction between event similarity and penalty **46**

Empirical prediction 2 – recall policy is sensitive to memory confusability and the cost of mistakes 47

Simulation 4: A familiarity signal can guide episodic recall policy **48**

Simulation 5: Encoding selectively at event boundaries benefits subsequent recall **51**

Empirical prediction 3 – the influence of encoding timing on subsequent recall 56

Simulation 6: Recall fills the gaps of the ongoing situation model **57**

Simulation 7: The cortical network integrates information over long timescales, even without the hippocampus **60**

Simulation 8: Integrating information across multiple event segments **63**

Simulation 9: Schema-consistent behavior when there is a prototypical event **68**

Empirical prediction 4 – the influence of schema strength and penalty on recall policy . . 76

General discussion **76**

Relation to other models . . . 80
Memory-augmented neural networks with a differentiable neural dictionary . . . 80
The structured event memory (SEM) model . . . 82
Future directions and limitations . . . 83
Extensions to the modeling framework . . . 83
Summary . . . 87

Acknowledgements **87**

Code **88**

Appendices **89**
Methods detail - Inter-subject correlation . . . 89
Model parameters . . . 90
Parameters of the task environment . . . 90
Parameters specific to particular simulations . . . 91

References **91**

List of Figures

1 The task conditions used in the experiment by J. Chen et al. (2016) . . . 9
2 Model architecture . . . 13
3 The model's internal representation during event prediction. . . . 19
4 A toy demo - how the situation governs the event transition dynamics . . . 20
5 A simulated version of the task used by J. Chen et al. (2016) . . . 23
6 The stimulus for a single trial . . . 27
7 Using the shared response model to functionally align neural networks – figure from Lu et al. (2018) . . . 31
8 Example – decoded internal states over time . . . 37
9 Prediction performance during part 2 for the three conditions, as a function of penalty 38

10 Memory activation and input gate values for the three conditions, as a function of penalty . . . 40
11 The effects of uncertainty and penalty on recall policy interact . . . 43
12 The level of penalty at test influences the current recall policy . . . 45
13 The effect of event similarity and penalty on recall policy . . . 47
14 The familiarity signal can improve event prediction . . . 49
15 Recall is modulated by familiarity . . . 50
16 Empirically, the neural signature of episodic encoding is time-locked with event boundaries . . . 53
17 Models that selectively encode at event boundaries perform better . . . 54
18 Why encoding selectively at event boundaries might help subsequent recall . . . 55
19 Models that encode in the middle of an event have higher lure activation . . . 56
20 Proposed experiment design to test the influence of encoding timing on subsequent recall . . . 57
21 Episodic recall makes the activity dynamics between the DM and RM conditions more similar: model of results from J. Chen et al. (2016) . . . 59
22 Scrambling sensitivity for controls and patients: empirical results from Zuo et al. (2020) . . . 61
23 Scrambling sensitivity without the hippocampus: simulation results for Zuo et al. (2020) . . . 62
24 The storyline effect from Chang et al. (2020): data and simulation . . . 65
25 Working memory in later blocks contains a more complete situation model . . . 67
26 Event graph with prototypical events. . . . 69
27 The model shows a schema-consistent false memory effect . . . 71
28 The model makes schema-consistent predictions when it is uncertain . . . 73
29 Input gate as a function of schema strength and penalty, plotted separately for schematic and non-schematic time points . . . 75
30 Inter-subject correlation . . . 89

In a natural setting, when should an intelligent agent encode and retrieve episodic memories? For example, suppose I am viewing the BBC television series *Sherlock*. Should I recall an episodic memory that I formed when I watched earlier parts of the show, and if so, when should I recall? When should I encode information about the ongoing episode?

Although episodic memory is one of the most studied topics in cognitive psychology and cognitive neuroscience, the answers to these questions are still lacking, in large part because episodic memory research has traditionally focused on experiments using simple, well-controlled stimuli, where participants receive clear instructions about when to encode and retrieve. For example, a typical episodic memory experiment could ask participants to remember a set of random word-pairs (Kahana, 2012). Later on, given a word-cue, the participants need to report the associated word (Kahana, 2012). In this word-pair experiment, the optimal timing for encoding and retrieval is clear: The participant should encode an episodic memory when they study a word-pair and retrieve the associate when they are prompted by a cue.

Recently, there has been increasing interest in using naturalistic stimuli such as movies or audio narratives in psychological experiments, to complement results from traditional experiments using simple and well-controlled stimuli (Nastase et al., 2020; Sonkusare et al., 2019). These experiments have the potential to shed light on when encoding and retrieval take place during event perception in a naturalistic context, where no one is explicitly instructing participants about how to use episodic memory. However, naturalistic paradigms of this sort are "out of scope" for existing computational models of human memory, which tend to focus on discretized list-learning paradigms like the (hypothetical) word-pair learning study described above – these models (see Norman et al., 2008 for a review) are designed to answer questions about what happens as a result of a particular sequence of encoding and retrieval trials, not questions about when encoding and retrieval should occur in the first place.

The goal of the present work is to extend the scope of memory models to naturalistic tasks where there is no explicit demand to use episodic memory, and to use the model to answer questions about how episodic memory actually gets used in these tasks. We formalize the task of event cognition by assuming that event sequences are samples from some underlying event schema, and that the agent's goal is to predict upcoming events. Then we seek to identify good

policies for event prediction and episodic encoding/retrieval by optimizing neural network agents on the tasks of interest. We can compare the policies that emerge from neural network training to human empirical data. To the extent that the learned policies capture human empirical data, the model can be viewed as providing a rational account of those findings.

More specifically, our modeling work leverages recent advances in memory-augmented neural networks (Graves et al., 2016; Ritter et al., 2018), deep reinforcement learning (Mnih et al., 2016; Sutton & Barto, 2018), and meta-learning (Botvinick et al., 2019; J. X. Wang et al., 2018) – these advances (collectively) make it possible for neural network models to *learn to use episodic memory* in the service of event prediction. The model has two parts: The first part (meant to simulate cortex) is a recurrent neural network (RNN) that integrates information over time and makes predictions about upcoming events. The model also has an episodic memory module (meant to simulate hippocampus) that stores snapshots of cortical patterns and reinstates these patterns to the cortical network. Using a reinforcement learning algorithm, the RNN learns a policy that controls the episodic memory module, thereby enabling the model to dynamically decide when to recall. In this paper, we analyze how the learned policy changes under different environmental regimes, and how well the learned policy captures human behavioral and neuroimaging data.

Modeling targets

In this section, we summarize two empirical results to provide concrete examples of modeling targets related to the questions of when to recall and when to encode. Both of these questions are critical for effective use of episodic memory in event cognition.

When to recall

Recent studies using naturalistic stimuli have started to shed light on how the brain deploys episodic memory during naturalistic movie viewing. In one experiment, participants viewed a TV episode from *The Twilight Zone* (J. Chen et al., 2016). Figure 1 illustrates the task conditions in the experiment. This episode was divided into two parts (part one and part two). Participants in the recent memory (RM) condition viewed the two parts back-to-back;

Figure 1

The task conditions used in the experiment by J. Chen et al. (2016). Reprinted with permission. Participants viewed a movie from the TV series *The Twilight Zone*. The movie was divided into two parts. Information from part one was useful for understanding part two. In the recent memory (RM) condition, the participants viewed the entire movie (part one and part two) continuously. In the distant memory (DM) condition, the participants also viewed the entire movie, but there was a one-day gap between viewing part one and part two. Consequently, recalling episodic memories from part one was useful for understanding part two. In the no memory (NM) condition, the participants only viewed part two.

participants in the distant memory (DM) condition had a one-day gap in between the two parts of this TV episode; participants in the no memory (NM) condition only watched the second part (J. Chen et al., 2016). In the RM condition, participants can build up a situation model – i.e., a representation of the relevant features of the ongoing situation (Richmond & Zacks, 2017; Stawarczyk et al., 2019; Zacks, 2020) – during the first part of the movie and actively maintain it over time; all of that information is still actively represented at the start of part two, so there is no need to retrieve it from episodic memory. By contrast, in the DM condition, a day has passed between part one and part two, so participants are no longer actively maintaining the relevant situation model at the start of part two; to recover this situation model (without having to wait to re-observe the features of the situation), participants need to retrieve it from episodic memory.

In the NM condition, participants do not have a stored situation model, so they are forced to construct one on the fly.

Importantly, people's interpretations of part two will be different depending on whether they are actively maintaining information from part one. For example, during part one, participants learned that all of the servants in the house are robots. In part two, one character in the story wants to kill these servants, and the fact that these servants are robots is not mentioned until five minutes later. Therefore, in the DM condition, retrieving episodic memories about part one is important for event understanding, especially at the beginning of part two.

In the study, the researchers found strong hippocampal-cortical activity coupling measured using inter-subject functional connectivity (ISFC; Simony et al., 2016) for DM participants during the first two minutes of part two, while the level of coupling was much weaker for participants in the RM and NM conditions (J. Chen et al., 2016). Notably, cortical regions that had a strong coupling with the hippocampus (in the DM condition) largely overlapped with the default mode network (DMN), which is believed to actively maintain a situation model (Stawarczyk et al., 2019). The study also found that – at the start of part two – inter-subject correlation (ISC) in DMN regions was lower between participants in the DM and RM conditions than between RM participants, suggesting differences in how DM and RM participants were interpreting the story. However, this gap in ISC decreased over the course of part two, suggesting that these differences in interpretation between DM and RM participants are reduced over time. Furthermore, across participants, the degree to which the gap in ISC narrowed during the second half of part two was correlated with the amount of hippocampal-cortical activity coupling at the start of part two (J. Chen et al., 2016). Taken together, these findings can be interpreted as showing that hippocampus is consulted more (as evidenced by increased hippocampal-cortical coupling) in the DM condition (where there are "gaps" in the situation model at the start of part two) than the RM condition (where the situation model is more complete); the effect of this increased consultation of the hippocampus is to "fill in the gaps" and align the interpretations of the DM and RM participants (as evidenced by DM-RM ISC rising to the level of RM-RM ISC). We model this detailed pattern of results in *Simulation 6*.

For now, the key take-away point from the J. Chen et al. (2016) study is that

hippocampally-based recall appears to occur more frequently in situations where there is a high *demand* for recall (i.e., where there are gaps in the participant's situation model that need to be filled to reduce uncertainty) compared to situations where there is a lower demand for recall (because the participant's situation model is relatively complete). Importantly, participants' experience in part one of the DM and RM conditions was exactly the same, so presumably they stored the same episodic memories during part one; despite this equivalence in stored memories, and also in the sensory cues available during part two (both groups were watching the same movie), it appears that participants in the "high recall demand" (DM) condition were drawing more strongly on the hippocampus. This "demand sensitive" view of hippocampus implies that the brain can be strategic in how it calls upon the hippocampus to support event understanding, and it underlines the importance of the aforementioned goal of specifying the exact policy for when retrieval should occur.

When to encode

The J. Chen et al. (2016) study speaks to when retrieval takes place during naturalistic stimulus processing – now we consider the complementary question of when episodic encoding occurs during naturalistic stimulus processing. To answer this question, several recent fMRI studies have measured hippocampal activity during perception of events and related this to later memory for the events. Specifically, Ben-Yakov and Dudai (2011) and Ben-Yakov et al. (2013) found that the end of a video clip triggered a hippocampal response, and the magnitude of the hippocampal response predicted later memory for the clip. One possible interpretation of this result is that, when an event ends, the hippocampus stores a "snapshot" of the event so you can remember it later. The aforementioned studies used unrelated video clips with an obvious ending; more recently, studies have found that this hippocampal response peak also occurs at event boundaries when participants watch continuous narrative movies (where the boundaries were annotated by a separate group of human subjects; Baldassano et al., 2017; Ben-Yakov and Henson, 2018; Reagh et al., 2020). Furthermore, Baldassano et al. (2017) found that the level of hippocampal response was correlated with the level of pattern reinstatement at recall in several cortical regions, including angular gyrus and posterior cingulate cortex. Taken together, these

results suggest that event boundaries might be a particularly important time for episodic encoding. However, existing studies do not offer a computational-level (i.e., normative) account of why the brain should preferentially encode episodic memories at event boundaries. Several researchers have argued that the current situation model is "flushed out" at event boundaries (Radvansky et al., 2011; Richmond and Zacks, 2017; for neural evidence in support of this dynamic, see Chien and Honey, 2020; Ezzyat and Davachi, 2011) – it is therefore relatively intuitive that encoding should occur before this "flushing out" occurs. However, it is less clear why encoding episodic memories within an event (i.e., well before the boundary) might be suboptimal. In the present study, we aim to provide an account of why selectively encoding at event boundaries is desirable (see *Simulation 5* below).

A model of cortical-hippocampal interaction

As noted above, our model is inspired by recent progress on memory-augmented neural networks (Graves et al., 2016; Ritter et al., 2018; Wayne et al., 2018) and deep reinforcement learning (Mnih et al., 2016) and related works in neural and cognitive modeling (Botvinick et al., 2019; J. X. Wang et al., 2018). The model has two main components that conceptually correspond to the cortical network (Figure 2A, shown in gray) and the hippocampus (Figure 2A, shown in blue).

The cortical network

The cortical network is a Long-Short-Term Memory network (LSTM; Hochreiter and Schmidhuber, 1997), which is a recurrent neural network (RNN) with gating mechanisms. The LSTM architecture gives the cortical network the ability to actively maintain and integrate information over time. For terminological convenience, we will refer to this active maintenance ability in the paper as "working memory". However, we should emphasize that – contrary to classic views of working memory (e.g., Baddeley, 2000) – our model does not have a working memory buffer that is set apart from other parts of the model that do stimulus processing; rather, active maintenance and integration are accomplished via recurrent activity in the parts of the model that are doing stimulus processing. In this respect, the architecture of our model fits with

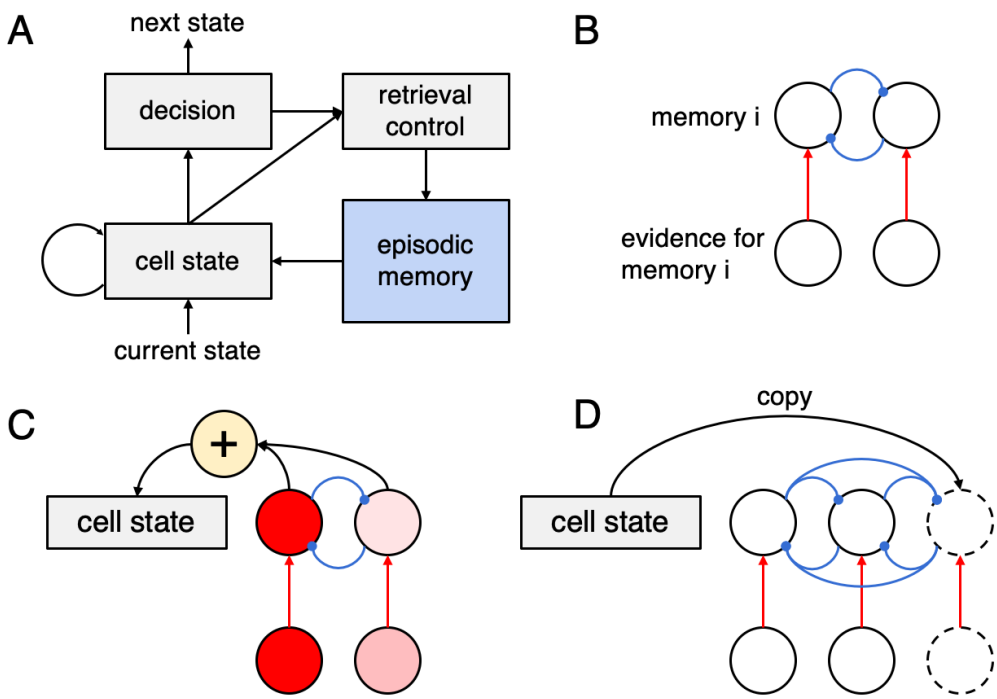

Figure 2

Model architecture. A) The model has several components. The cell state is a Long-Short-Term Memory (LSTM) layer that accumulates information over time. The decision layer outputs the predicted upcoming event. The episodic memory module stores a set of memories (previously saved cell states), which can be reinstated to the current cell state. The retrieval control layer sets the input gate value of the episodic memory module. **B)** Episodic retrieval is an evidence accumulation process, which takes the evidence for each memory and computes the activation, or weight, for each memory. Then it returns a weighted average of all memories. The activation of a memory is positively related to the evidence for that memory – the cosine similarity between the memory pattern and the current LSTM cell state. The setting of the input gate (red arrows) has a multiplicative influence on the evidence – a higher input gate value means more recall. The level of competition (blue edges) scales the level of lateral inhibition across different memories. **C)** Once the activation value is computed for each memory, the weighted average pattern across all memories is reinstated back to the cell state. In this example, the first (left) memory received more evidence and became strongly activated, which led to a suppression of other memories (that received less evidence). **D)** Encoding an episodic memory corresponds to adding a new evidence accumulator to the model.

the *process memory* framework set forth by Hasson et al. (2015).

In the context of event processing, this cortical network mainly corresponds to the default mode network (DMN) in the brain. It has been proposed that, during event processing, the DMN actively maintains a situation model – a predictive model that maintains information about the relevant features of the ongoing situation and the transition structure across events (Stawarczyk et al., 2019; Zacks, 2020). In our model, the ongoing situation model is represented by a pattern of neural activity across the hidden layer (cell state) in the RNN.

The hippocampus

An episodic memory module (Figure 2A, shown in blue) is attached to the cortical network; this module functionally represents the hippocampus. The episodic memory system stores a set of memories, which are represented as a set of mutually competing evidence accumulators (Figure 2B). Each memory trace indexes a previously saved cortical pattern; the extent to which memories are activated at retrieval is a function of the match between the previously encoded cortical pattern versus the current cortical pattern. The retrieval process takes a linear combination of episodic memories, weighted by their activation, and adds the retrieved pattern to the cell state of the cortical network (Figure 2C).

Episodic encoding

To encode an episodic memory, the model takes a snapshot of the pattern of neural activity in the cortical network. Specifically, episodic encoding corresponds to adding a new evidence accumulator corresponding to the current LSTM cell state pattern to the set of existing episodic memories (Figure 2D). Since encoding a new memory does not modify any existing memories and the new memory is stored independently from others, this implementation of encoding instantiates pattern separation – a process that assigns relatively non-overlapping hippocampal patterns to potentially overlapping cortical patterns to reduce interference (O'Reilly & McClelland, 1994; Rolls, 2013; Yassa & Stark, 2011).

In most experiments in this paper, the model selectively encodes episodic memories at event boundaries (i.e., at the end of an event) – this assumption is motivated by human

neuroimaging results showing that hippocampal activation peaks at event boundaries (Baldassano et al., 2017; Ben-Yakov & Henson, 2018) and that the size of the hippocampal peak predicts subsequent recall of the just-completed event (Baldassano et al., 2017; Reagh et al., 2020). More importantly, when we experimented with alternative encoding regimes, we found that selectively encoding at event boundaries leads to the best performance in a subsequent task that is dependent on episodic memory (*Simulation 5*). Note that, since the cortical network can actively maintain features in working memory after observing them, taking an episodic "snapshot" of the model's working memory state at a particular moment in time can result in the storage of multiple features of the present situation (i.e., all of the features currently maintained in working memory).

Episodic recall

Episodic retrieval is content-based. This process returns a weighted average of all episodic memories; to calculate the weight, or the activation, for each memory, the model executes an evidence accumulation process. The evidence for a given episodic memory is the similarity between that memory versus the current cortical pattern in the hidden layer (cell state). Hence, memories that are similar to the current cortical pattern will have a bigger influence on the pattern that gets reinstated. This conceptualization of episodic memory is similar to an attractor network (Hopfield, 1982; Rolls, 2010) – each episodic memory serves as an attractor in the space of LSTM cell states, and retrieval simply moves the LSTM cell state towards those episodic memories.

The evidence accumulation process is governed by the input gate (in red, Figure 2B) and the level of competition across memories (in blue, Figure 2B). The input gate is controlled by the cortical network via the retrieval control layer (Figure 2A) and applied uniformly to all memories. The input gate adjusts the influence of episodic recall – a higher input gate increases the activation for all the memories. The level of competition, or lateral inhibition, acts like a soft-max temperature, which adjusts the contrast of activations across all memories; making the level of competition higher or lower interpolates between one-winner-take-all recall versus recalling an average of multiple memories. In all of our simulations, we chose the level of competition to be large given the overwhelming evidence that episodic recall is competitive (Anderson & Reder,

1999; Norman, 2010; Norman & O'Reilly, 2003).

Episodic recall – detail

The evidence accumulation process used in our model is a simplification of the leaky competing accumulation (LCA) process (Usher & McClelland, 2001). The LCA is closely related to the drift diffusion model for decision making (Bogacz et al., 2006), and it has been used in several other memory models (e.g., Polyn et al., 2009; Sederberg et al., 2008). Conceptually, our model uses an LCA process that 1) has no leak; 2) has no noise; 3) uses the same level of input gate for all accumulators; 4) uses the same level of lateral competition for all accumulators. Additionally, instead of optimizing the LCA parameters to fit empirical results (e.g., as in the work by Polyn et al., 2009), we use a neural network that learns to control the level of the input gate value. As described below, in the *Objective function and learning algorithm* section, the model's goal is to maximize reward by making correct predictions and avoiding incorrect predictions; the network learns a policy for setting the input gate that maximizes the reward it receives.

At time t, assume the model has n memories. The model first computes the evidence for all of the memories. The evidence for the i-th memory, m_i, is the cosine similarity between the current LSTM cell state pattern versus that episodic memory – which is a previously saved LSTM cell state pattern. We denote the evidence for the i-th memory as x_i:

$$x_i = \text{cosine}\left(c_t, m_i\right)$$

The x_i, for all i, are the input to the evidence accumulation (LCA) process used in our model; the evidence accumulation process has a timescale τ that is faster than t, such that the accumulation process runs to completion within a single timestep of the cortical model. The computation at time τ (for $\tau > 0$) is governed by the following formula:

$$w_\tau^i = \text{relu}\left(\alpha x_i - \beta \sum_{j \neq i} w_\tau^j\right)$$

w_τ^i is the activation value for the i-th memory at time τ. The activation for the i-th memory is positively related to its evidence, x_i, and is multiplicatively modulated by α, the input

gate. The i-th memory also receives inhibition from all of the other memories, where the level of inhibition is modulated by the level of competition, β. Finally, the retrieved item at time t, denoted by μ_t, is a combination of all memories, weighted by their activation:

$$\mu_t = \sum_{i=1}^{n} w_i m_i$$

Objective function and learning algorithm

The model is trained with reinforcement learning. Specifically, the model is rewarded/penalized if its prediction about the next event is correct/incorrect. The model also has the option of saying "don't know" (implemented as a dedicated output unit) when it is uncertain about what will happen next; if the model says "don't know", the reward signal is zero. In the real world, there are different penalties associated with making commission errors (wrong predictions) and omission errors (not making a prediction). Having the "don't know" option gives the model the freedom to choose whether it should make a specific prediction (thereby incurring the risk of making a commission error and receiving a penalty) or whether it should express uncertainty to avoid a possible penalty. Intuitively, this choice should depend on the environment. For example, if the penalty for misprediction is zero, the model should make an event prediction even if it has high uncertainty about the upcoming event. In contrast, if the penalty for misprediction is high, the model should only make an event prediction if it is certain about what would happen next.

The model is trained with the advantage actor-critic (A2C) objective (Mnih et al., 2016). At time t, the model outputs its prediction about the next state (i.e., the next event), $\hat{s_{t+1}}$, and an estimate of the state value, v_t. After every sequence of events (of length T), it takes the following policy gradient step to adjust the connection weights for all layers, denoted by θ:

$$\nabla_\theta J(\theta) = \nabla \sum_{t=0}^{T} \log \pi_\theta \left(\hat{s}_{t+1} | s_t\right) \left(r_t - v_t\right)$$

Intuitively, this objective makes rewarded actions (event predictions) more likely to occur. Additionally, this process is also modulated by the level of reward prediction error – measured as the difference between predicted value, v_t, versus the reward at time t, denoted by r_t. We also used entropy regularization on the network output (Grandvalet & Bengio, 2006; Mnih et al.,

2016) to encourage exploration in the early phase of the training process.

We used the A2C method (Mnih et al., 2016) mainly for its simplicity. A2C has been widely used in cognitive modeling (Ritter et al., 2018; J. X. Wang et al., 2018). Notably, there is also evidence that an actor-critic style system is implemented in the cortex and basal ganglia (Takahashi et al., 2008).

Model execution

Figure 2A shows the model architecture. At time t, the model takes the current event and predicts the next event. Internally, the LSTM integrates the input with its cell state from the previous time step and passes the output to the decision layer. Then the control layer computes the input gate value based on the LSTM cell state and the activity of the decision layer. The input gate value and the level of competition modulate episodic recall, which returns an episodic memory (a pattern of neural activity) to the LSTM layer. This influences the decision layer again. Finally, the decision layer generates a prediction for the upcoming event.

The decision layer, the non-linear layer on top of the LSTM (see Figure 2A), is useful for the model to learn the task. This is because evaluating internal uncertainty, which is needed to compute the "don't know" response, makes the event prediction problem more non-linear. To see this (Figure 3), we projected the patterns of neural activity from the decision layer (when the model is making an event prediction) onto the subspace spanned by the first two principal components. Then we colored these patterns according to the model's final choice/predicted event. First, we found that the activity patterns for different choices occupy different regions in the state space, such that it is possible to use a linear decision boundary to separate one choice versus the rest of the choices (Figure 3A,B). Moreover, the states of uncertainty tend to cluster in the origin of the state space, which happens to be the center of all the choices (Figure 3A). Mathematically, the model appears to use the norm of the activity pattern vector to encode the level of confidence, and the origin of the state space is the region of uncertainty (Figure 3C). This pattern is also observed in other studies (Dhamija et al., 2018). A cognitive-level interpretation is that uncertainty is represented by the absence of evidence of any particular choice.

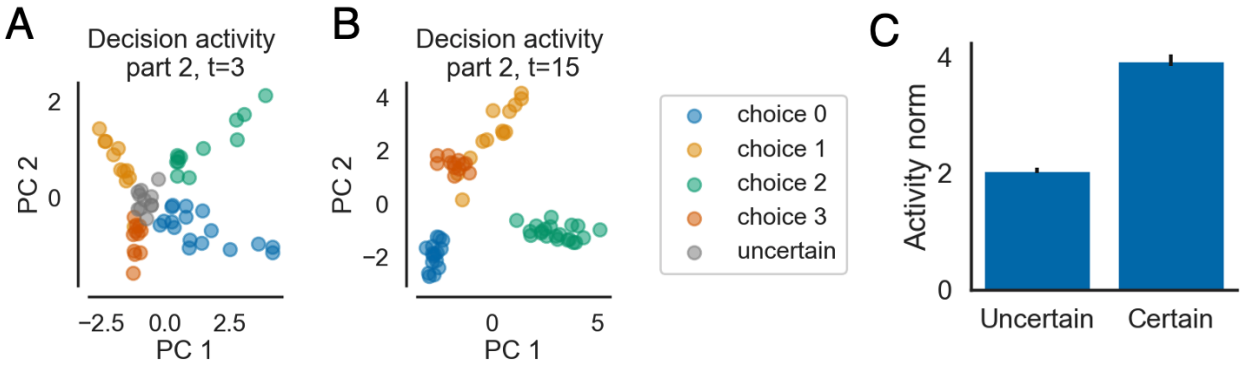

Figure 3

The model's internal representation during event prediction. Parts A and B show the neural activity patterns from the decision layer, projected onto the subspace spanned by the first two principal components. Each point corresponds to the pattern of neural activity for a trial at a particular time point. We colored the points based on the output (i.e., "choice") of the model, which represents the model's belief about which event will happen next. Patterns that subsequently led to "don't know" responses are colored in grey. The two panels (A and B) show two example time points from a trained model during part two in the distant memory (DM) condition (see experiment design in the next section). Panel A shows an early time point with substantial uncertainty (a large number of "don't know" responses). Panel B shows the last time point of this event sequence, where the model has lower uncertainty. Note that uncertain states are approximately clustered in the origin of the state space while other responses are farther away, which indicates that uncertainty in our model is represented by the absence of evidence towards any particular choice. Panel C shows the average L2 norm of states that led to don't know responses (uncertain) versus states that led to event predictions (certain). We found that internal states that subsequently led to "don't know" responses are closer to the origin.

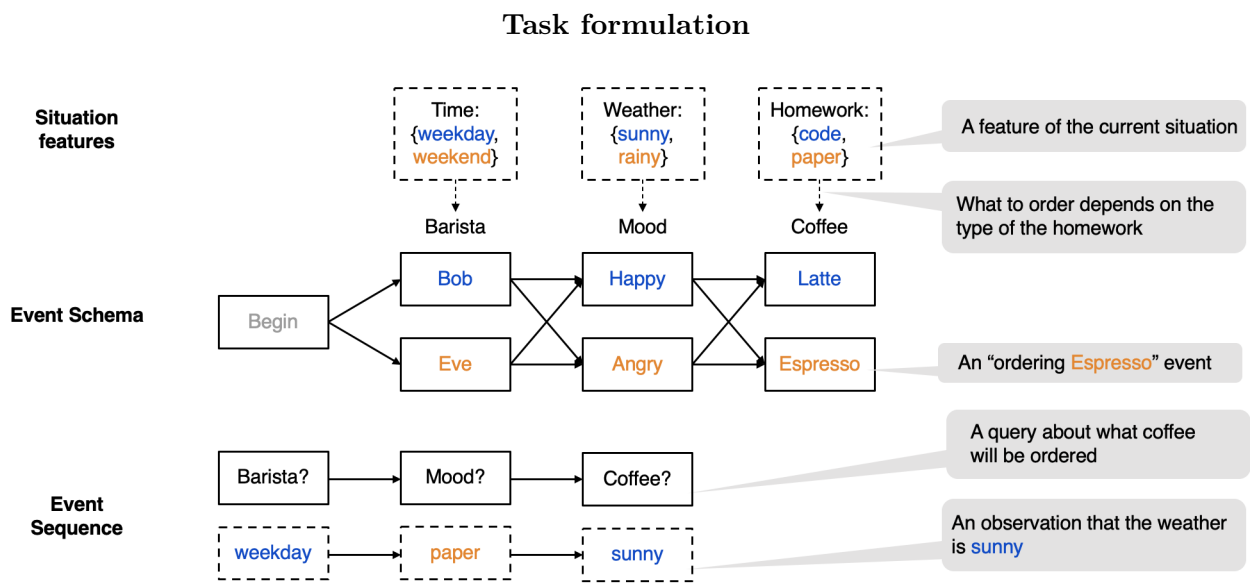

Figure 4

A toy demo - how the situation governs the event transition dynamics. When the model is processing an event sequence, at time t, the model observes a situation feature and gets a query about the upcoming event. An event sequence is a sample from its underlying event schema (a directed graph). Importantly, the transition structure of the schema is controlled by the ongoing situation, specified by a set of situation features. For example, the time of the day (weekday or weekend) controls the identity of the barista (Bob or Eve) in this event sequence. We can also use this process to generate stimuli for human behavioral experiments (see the proposed study in *Empirical Prediction 1*).

An event sequence is a sample from an event schema conditioned on a situation

Figure 4 provides a high-level illustration of the process by which stimuli are generated. We define an event schema as a graph, where each node is an event, and each edge is a transition between two events. The set of all edges defines the transition matrix across events. When we take a sample from this graph, we get an event sequence. The schema graph has a (horizontal) length of T, so the event sequence will contain T events (in Figure 4, $T = 3$, since we discount the start node). The number of nodes in each layer (column) of the schema graph determines the number of possible events for each time step (in Figure 4, there are 2 possible events for each time

step, but in our actual simulations there are 4 possible events for each time step).

Importantly, transitions are not random – rather, they are conditioned on a list of feature-value pairs that we refer to as the current *situation*. Concretely, the transitions leading into each time point (column) of the graph are deterministically controlled by the value of a particular situation feature. To illustrate this point, consider the example shown in Figure 4. The event graph represents a coffee shop schema. When sampling a path from this schema, we obtain a story with three events – meeting the barista, learning about their mood, then ordering a cup of coffee. The three transitions in this schema are controlled by three situation features – time of the week, the weather and the homework, respectively. For example, at the first time step, the particular "meeting the barista" node that is visited depends on the time of the week – Bob is the barista on weekdays, and Eve is the barista during the weekend; and at the second time step, the "barista's mood" node that is visited depends on the weather – the barista is happy when it is sunny and angry when it is rainy.

When processing an event sequence, at time t, the model first gets to observe a situation feature value (e.g., time = "weekday"), then the model is asked to predict what will happen next. Because each event transition is conditioned on a particular feature of the ongoing situation (e.g., the weather), asking the model to predict the next node can be viewed as a query to the feature that controls that transition. If the model knows the queried feature, then it can predict the next event correctly.

A key property of the event sequences is that, while the order of queries about what will happen next is fixed (e.g., you always are asked to predict if the barista is Bob or Eve before being asked to predict if they are happy or angry), the order in which situation features are observed by the model is random. For example, for some coffee shop sequences, the model will observe the time-of-week then the homework-type then the weather, but in other sequences the model will observe weather then time-of-week then homework-type. Since the number of situation features and the number of queries are the same, this means that around 50% of the time the model will observe the situation feature that controls a transition before it is asked to predict that transition, and around 50% of the time the model will observe that feature after it is asked to predict that transition. In the former case (where the feature is observed before the transition it

controls) the model should be able to correctly predict the upcoming event, so long as it has the feature stored in its working memory (note that the interval between observing a feature and being asked to make a prediction based on that feature can be quite long, which incentivizes the model to use its LSTM cell state to maintain features over time). In the latter case (where the feature is observed after the transition it controls), the only way that the model can respond correctly (other than guessing) is if it retrieves that feature from episodic memory.

To pull this all together, consider the example event sequence in Figure 4. At time 1, the model observes the time of the week, then it gets queried about the identity of the barista. Since the time of the week determines the identity of the barista, the model should be able to make the correct prediction based on its working memory. At time 2, the model is queried about the barista's mood, which depends on the weather. Since the weather has not yet been observed, the model may not have the information required to predict the upcoming event in working memory. However, even though the model has not observed the weather during this particular sequence, it might have a stored episodic memory of doing coding homework on a weekday when it is sunny; based on the model's prior observation (at time 1) that it is a weekday, it might retrieve the coding-weekday-sunny episodic memory, thereby loading "sunny" back up into working memory (even though it was not directly observed) and – based on this – the model might make a prediction that the barista will be happy.

Task conditions

To delineate the contributions of working memory and episodic memory, we adopted the task conditions used in J. Chen et al. (2016): recent memory (RM), distant memory (DM), and no memory (NM) (Figure 1). A diagram of the task procedure is shown in Figure 5. For all conditions, the model starts by viewing an event sequence sampled from situation A (a1; Figure 5). The model's working memory is then flushed by resetting the activity of the LSTM to the zero vector; this is meant to capture the passage of time between viewing the first event sequence and subsequent sequences. Then, for the RM and the DM conditions, the model sees two event sequences (b1 and b2, Figure 5) from situation B (shown in yellow, Figure 5). These two event sequences with a shared situation mimic the two-part movies used in the study by J. Chen et al.

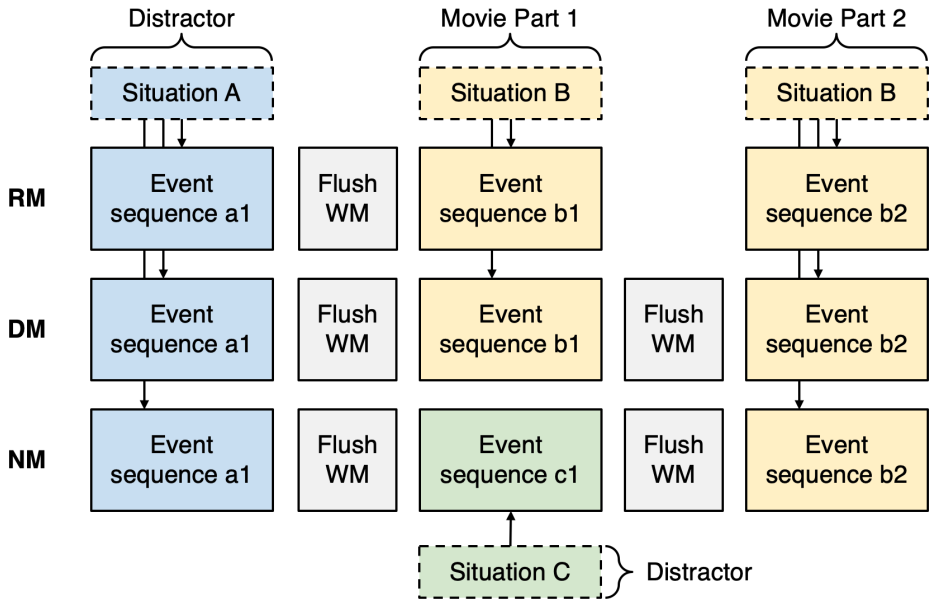

Figure 5

A simulated version of the task used by J. Chen et al. (2016). Each row represents a trial in a particular condition. The model stores an episodic memory at the end of each event sequence (see text). In the RM and the DM conditions, the model first sees two event sequences a1 and b1, generated by situation A and situation B. Then situation B occurs again, which generates event sequence b2. Though b1 and b2 are different in terms of the order in which situation features are presented, all situation features agree, since both sequences were generated by situation B. The only difference between the RM and DM conditions is that, in the RM condition, the model has relevant information about the ongoing situation (i.e., situation B) stored in working memory at the start of part two, whereas the model in the DM condition does not, because working memory is flushed between b1 and b2. In the NM condition, the model sees part two (b2) without having any information about situation B stored in working memory or episodic memory (because it has not previously encountered any sequences from situation B).

(2016). Although these two event sequences (b1 and b2) differ in terms of the order in which situation features are presented, the situation features have the same values, because they were generated by the same situation (situation B). Therefore, recalling an episodic memory about part one can inform event prediction during part two. We were mainly interested in the model's behavior during part two of the sequence (b2). The model encodes episodic memories at the end of each event sequence (i.e. at event boundaries), since we found this is the optimal encoding policy (*Simulation 5*) and this is consistent with empirical results (Baldassano et al., 2017; Ben-Yakov & Henson, 2018; Reagh et al., 2020). During part two (b2), the model has two stored episodic memories to draw upon: the episodic memory formed at the end of the initial B sequence (b1), which we refer to as the *target* memory (since it pertains to the same situation, and thus recalling this memory will help prediction), and the episodic memory formed at the end of the A sequence (a1), which we refer to as the *lure* memory (since it pertains to a different situation, and thus recalling this memory will hinder prediction). By design, target-lure similarity is controlled by the similarity between the two situations (situation A and B). Because we were primarily interested in how the model learns to use episodic memory during part two of the sequence, we blocked the model from using episodic memory (and from updating its gating policy for episodic memory) during part one of the sequence. [1]

During part two, in the RM condition, the model has information about the ongoing situation stored in both episodic and working memory. In the DM condition, the model's working memory is flushed in between the two parts (b1 and b2). This simulates the effect of having a one-day gap in between part one and part two in the experiment by J. Chen et al. (2016). In the DM case, during part two, the model only has information about the ongoing situation stored in episodic memory (not working memory, since this was flushed). Finally, in the NM condition, the model sees three event sequences from three different situations. This is similar to seeing part two

[1] Episodic memory is not useful for making predictions during part one of the sequence, because part one is always a new, never-before-seen situation (see Figure 5). Thus, allowing part one of the sequence to influence recall policy biases the model to simply not use episodic recall. One way around this issue would be to simulate more complex scenarios where the "base rate" availability of relevant episodic memories is higher, thereby encouraging the model to rely more on episodic recall. However, the model already takes a long time to train in the current (simplified) environment, so we opted not to explore these more complex scenarios in this paper.

of the movie without seeing part one in the study by J. Chen et al. (2016); in this condition, the model does not have relevant information about the ongoing situation (b2) stored in working memory or episodic memory at the start of part two.

Model training

Before the model is used to simulate any particular experiment, it undergoes a *model training* phase that is meant to reflect the experience that a person has prior to the experiment. The goal of this model training phase is to let the model learn 1) the structure of the task – how situation features control the transition dynamics across events; and 2) a policy for retrieving episodic memories and for making event predictions that maximizes the reward it receives. For every epoch of model training, it is trained for all three conditions (recent memory, distant memory, and no memory). The optimization objective is to maximize the model's performance on predicting the upcoming event (see *Objective function and learning algorithm* section).

Since pure reinforcement learning is not data-efficient enough, supervised initialization is used during training (Misra et al., 2017; Nagabandi et al., 2017). Specifically, the model is first trained for 600 epochs to predict the next event and minimize the cross-entropy loss between the output and the target. During this supervised pre-training phase, the model is only trained on the recent memory condition and the episodic memory module is turned off, so this supervised pre-training does not influence the development of the network's retrieval policy. Additionally, the "don't know" output unit is not trained during the supervised pre-training phase (we did this so the model could learn its own policy for saying "don't know", rather than having one imposed by us). Next, the model is switched to the advantage actor-critic objective (Mnih et al., 2016) and trained for another 400 epochs, allowing all weights to be adjusted.

During test (i.e., when simulating a particular experiment), the weights of the cortical model are frozen, but the model is allowed to form new episodic memories. This instantiates the assumption that cortex learns slowly and thus very little cortical plasticity will occur within a single experimental session (McClelland et al., 1995). Specifically, in any given trial where the model observes several event sequences, new learning of information completely relies on working memory in the model's recurrent dynamics, episodic memory in the episodic module, and

semantic memory encoded in the (frozen) cortical connection weights (instantiating the model's knowledge of transitions between events and how these transitions are controlled by situation features). The results shown in all of the simulations (below) were obtained by testing the model with new event sequences, after the initial training phase.

Stimulus representation

At time t, the model observes a situation feature, and it gets a query about what will happen next. Specifically, the input vector at time t has four components (see Figure 6): 1) The observed situation feature (sticking with the example in Figure 4, this could be something like "weather") is encoded as a T-dimensional one-hot vector. T is the total number of situation features, which (in most simulations) is the same as the number of time steps in the event sequence. The t-th one-hot indicates the situation feature governing the transition at time t. 2) The value of the observed situation feature (e.g., learning that the weather is sunny) is encoded as a B-dimensional one-hot vector, where B is the number of possible next events at time t. 3) The queried situation feature is encoded as another T-dimensional one-hot vector. 4) Finally, the model also receives the current penalty level on incorrect predictions as a scalar input, which can change across trials. Overall, the input vector at time t is $2T + B + 1$ dimensional. At every time point, there is also a target vector of length B that specifies the value of the queried feature (i.e., the "correct answer" that the model is trying to predict). The model outputs an event prediction vector of length $B + 1$: The first B dimensions represent possible values for the current situation feature, and the last output dimension corresponds to the "don't know" unit; as discussed earlier, if the model has not yet observed the situation feature that reveals what will happen next, the model has the option of saying "don't know" instead of guessing a specific value.

In our simulation, the length of an event sequence is 16 (i.e., $T = 16$), and the number of next events is 4 (i.e., $B = 4$). Hence the chance level for event prediction is 1/4. Figure 6 illustrates the stimuli provided to the model for a single example trial. Note that the queries (about the next event) are always presented in the same order, which captures the sequential structure of events (e.g., ordering food always happens before eating the food). However, the order in which the situation features are observed is random.

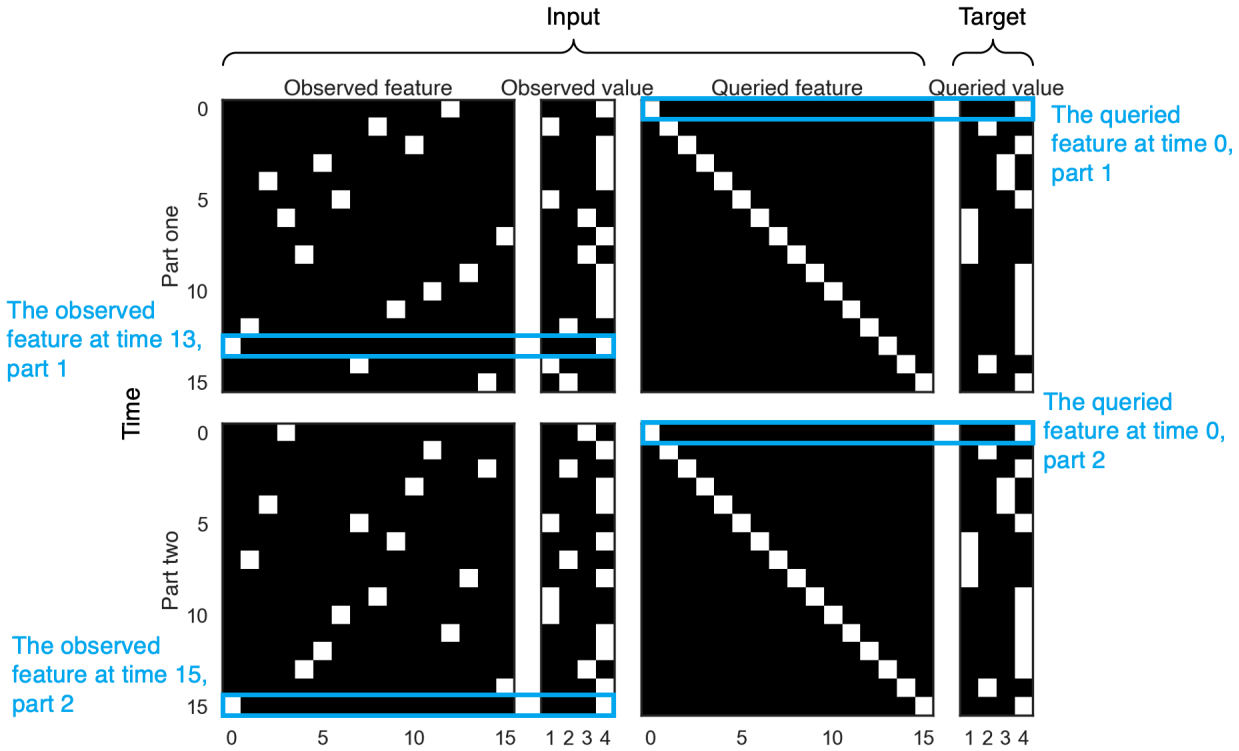

Figure 6

The stimulus for a single trial. Each row of the diagram shows the input vector and the target vector for the model at a particular time point. The columns of the diagram correspond to units in the network; when a unit is active at a particular time point, it is shown in white, otherwise it is shown in black. There are 32 rows total, corresponding to the 16 time points in part one of the sequence and the 16 time points in part two of the sequence. The input vector at time t consists of three "one-hot" parts: The first part specifies which feature (out of 16 possible features) is observed, the second part specifies the value of the observed feature (out of 4 possible values), and the third part specifies which feature is queried (out of 16 possible features). The additional input unit that encodes penalty information (see text) is not shown here. The target vector at time t specifies the value of the queried feature (out of 4 possible values). For example, at time 0, the first feature was queried (blue box in the figure), but the value of this feature was not observed until time 13 (i.e., long after it was queried). During part two, this feature was queried at the beginning and observed at the very end of the sequence. Note that the ordering of the observed features is random (i.e., it varies across part one and part two of the sequence) while the ordering of the queried features is fixed.

As noted above, the input vector specifies the level of penalty (for predicting the wrong event) for the current trial. Ideally, the model should behave more conservatively when the penalty is high. To examine this, we manipulated the distribution of penalty values to examine its effect on model behavior. In the simulations that follow, we trained some models in an environment where the penalty value was always zero. We also trained some models in an environment where the penalty value was always four. Models in the latter case should be more conservative, since they experienced higher penalty value on average (see *Simulation 1* for relevant results). Moreover, we trained some models that experienced a range of penalty values (from zero to four) during training; we then examined how these models behaved when the penalty was low (e.g. zero) versus high (e.g. four) during test, which allowed us to assess whether our model could adjust its policy in a trial-by-trial fashion (see *Simulation 2* for relevant results).

In most of the simulations, the model only got to observe 70% of the features of the ongoing situation during part one of the sequence (this was true during the simulated experiment and also during model training). This was operationalized by giving each feature a 30% probability of being removed during part one – for time points where the to-be-observed feature was removed, the model observed a zero vector instead. This "feature removal" during part one of the sequence made the task more realistic, since – in general – past information does not fully inform what will happen in the future.

Finally, we wanted to make sure the model could adjust its recall time flexibly, instead of learning to always recall at a fixed time step (e.g., always recall at the third time step). Therefore, during training, we delayed the prediction demand by a random number of time steps (from 0 to 3). For example, if the amount of delay was 2 in a given trial, then the model observed 2 situation features before it received the first query.

Analysis methods

Multivariate pattern analysis

We used a decoding approach to track what information the model was maintaining in working memory over time; this approach allowed us to assess the model's ability to hold on to observed features after they were observed, and also to detect when features were retrieved from

episodic memory and loaded back up into working memory. Our use of decoders here is analogous to the widespread use of multivariate pattern analysis (MVPA) methods to decode the internal states of participants from neuroimaging data (Haxby et al., 2001; Lewis-Peacock & Norman, 2014; Norman et al., 2006) – the only difference is that, here, we applied the decoder to the internal states of the model instead of applying it to brain data. This decoding approach was used to understand how working memory and episodic memory contributed to performance in *Simulations 1, 8, and 9*.

When using the decoder to gain insight into our "standard" paradigm (shown in Figure 5), we trained classifiers on LSTM cell states during part one to decode the feature values over time. Each situation feature was given its own classifier (logistic regression with L2 penalty). For example, if "weather" was one of the situation features, we would train a dedicated "weather" classifier that takes the LSTM cell state and predict the value of the weather feature for a given time point. To set up the targets for these classifiers, we labeled all time points as "don't know" before the model observed the feature value. After a feature value was revealed, we labeled that time point and the following time points with the value of that feature (e.g., if the weather feature value was observed to be "rainy" on time point 4, then time point 4 and all of those that followed until the end of part one of the sequence were labeled with the value "rainy"). This labeling procedure assumes that observed features are maintained in working memory but does not require this to be strictly true; the assumption can be tested by applying the classifier to held-out data. When decoding working memory states during part one of the sequence, we used a ten-fold cross-validation procedure on part one, and picked the regularization parameter with an inner-loop cross-validation; when decoding working memory states during part two, we trained the classifier on all of the part one data and tested the trained classifier on part two. Again, the regularization parameter was picked with an inner-loop cross-validation. All results were generated using held-out test sets. The average decoding accuracy values for part one and part two were 87.06% and 79.40%. Note that, as mentioned above, there is no guarantee that features observed earlier in the sequence will be maintained in the model's working memory. As such, below-ceiling decoding accuracy could reflect either 1) failure to accurately decode the contents of working memory, or 2) the decoder accurately detecting a working memory failure (i.e., that the

feature in question has "dropped out" of the model's working memory, despite having been observed earlier in the sequence).

Functionally aligning multiple neural networks

Inter-subject analyses are widely used in naturalistic studies, such as the studies by J. Chen et al. (2016) and Zuo et al. (2020) that we discuss in *Simulations 1, 6, and 7*. Comparing data across subjects requires some form of alignment. Human fMRI data are approximately aligned across subjects, since brain anatomy is highly similar across people. However, when many instantiations of the same neural network architecture are trained on the same data, they tend to acquire different neural representations, even though they represent highly similar mathematical functions (Dauphin et al., 2014; Li et al., 2015; Meng et al., 2018). That is, the same input can evoke uncorrelated neural responses across different networks, although they produce similar outputs. For our purpose, this means that directly correlating hidden-layer activity patterns across neural networks will underestimate the similarity of representations across networks. Therefore, to simulate effects involving (human) inter-subject analyses, we need a way to compare neural representation across neural networks.

To perform inter-subject analyses across neural networks, we functionally aligned neural networks using the shared response model (SRM) (Lu et al., 2018) – a functional alignment procedure commonly used for multi-subject neuroimaging data (P.-H. Chen et al., 2015; Haxby et al., 2011; Haxby et al., 2020). Intuitively, this method applies rigid body transformation to align different network activities into a common space (Figure 7). Previously, we have shown that neural networks with highly overlapping training experience can be aligned well with SRM (Lu et al., 2018). In *Simulations 6 and 7*, where inter-subject analyses are involved, we used SRM to align neural network activities. We used the Brain Imaging Analysis Kit (BrainIAK) to implement SRM (Kumar, Ellis, et al., 2020; Kumar, Michael Anderson, et al., 2020).

Brief summaries of simulations

In this section, we briefly summarize key findings from our simulations. All simulations use the task described in the *Task formulation* section with minor variations. Most of them

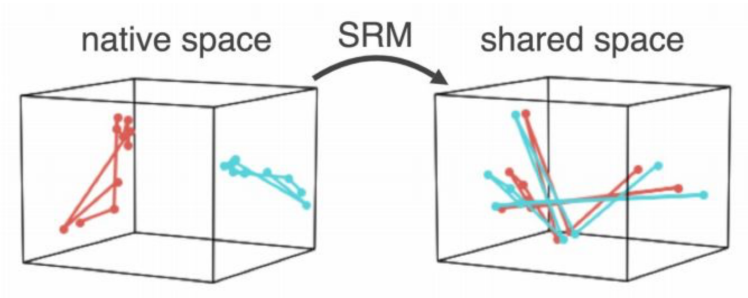

Figure 7

Using the shared response model to functionally align neural networks – figure from Lu et al. (2018). Reprinted with permission. In general, different neural networks with the same training set learn to approximate highly similar mathematical functions. However, since these models are overparameterized, their weights will be different from each other, so they will have different patterns of neural activity given the same input stimulus. This poses a challenge when conducting inter-subject analyses on neural networks. By contrast, inter-subject analyses on human brains are possible to some extent in the native voxel-space, since brains are roughly aligned due to anatomy (though it is known that both anatomical and function alignment can improve the quality of alignment; P.-H. Chen et al., 2015; Haxby et al., 2011). To perform inter-subject analyses (e.g., spatial and temporal inter-subject correlation) across neural networks, we used a functional alignment method – the shared response model (SRM) – to map the neural representations across different networks to a shared feature space. Our previous studies have demonstrated the validity of this approach (Lu et al., 2018).

include recent memory (RM), distant memory (DM), and no memory conditions (NM), as described in the previous section.

In *Simulation 1*, we start by showing that the model uses episodic memory to predict upcoming events. Specifically, we show that the model's performance in the DM condition is much better than in the NM condition; the only difference between these two conditions is that a correct episodic memory is available to the model in the DM but not the NM condition.

Second, we show that the model's learned recall policy is modulated by uncertainty: The model refrains from recalling when it thinks it knows what will happen next. This is illustrated by the model showing lower levels of recall in the RM condition (where the model has a complete

situation model maintained in working memory at the outset of part two) than in the DM condition (where the model's working memory is wiped between part one and part two). It is also illustrated by the model showing lower levels of recall in the DM condition when it has already observed the feature it is being queried about (and thus knows what will happen next) vs. when it has not already observed that feature. We explain these behaviors in terms of the idea that, when the model already understands what is going on, there is no benefit in recalling from episodic memory (there is no "gap" to fill), and there is potentially a cost (if it recalls a lure memory and overwrites the correct feature in working memory) – in this situation, the reward-maximizing policy is to wait to recall. This result leads to a novel experiment proposal, described in *Empirical Prediction 1*.

Third, we demonstrate that the learned recall policy is also modulated by the penalty on incorrect predictions. When the model is trained in an environment where the penalty (i.e., the cost of making an incorrect event prediction) is high, it waits longer to recall, relative to when the penalty at training is lower. This behavior can be explained in terms of a speed-accuracy trade-off: Waiting longer to recall allows the model to observe more features (disambiguating the present situation from other, similar situations, and thereby reducing the odds of false recall), but it also carries an opportunity cost (in terms of foregoing rewards it would have received if it had recalled correctly). Increasing the penalty causes the model to favor accuracy (i.e., avoiding costs of false recall) over speed.

In *Simulation 2*, we hold the distribution of penalty values at training constant (all models are trained on both low-penalty and high-penalty trials) and vary the penalty value at test. We show that the model can change its retrieval policy depending on the current trial's penalty value: When the penalty for the current trial is high, the model delays its recall time to make recall more accurate. We view this as a proof-of-concept demonstration that, rather than learning a single, context-insensitive policy during training, the model learns a "library" of policies, and it can flexibly access the correct policy depending on the prevailing conditions at test.

In *Simulation 3*, we simultaneously manipulate both the penalty for making incorrect event predictions and the average similarity of the target and distactor sequences. When these sequences are more similar on average, this increases the risk of recalling the lure memory. We

show that models trained in a high-event-similarity setting adapt to this by suppressing recall, especially when the penalty is high; by contrast, models trained in a low-event-similarity setting show high levels of recall regardless of the penalty – they can afford to do this because the memories are highly discriminable and thus the risk of false recall is low. This result leads to a novel experiment proposal, described in *Empirical Prediction 2*.

In *Simulation 4*, we provide a ground-truth familiarity signal that indicates whether the model has a stored episodic memory about the ongoing situation. We show that the learned recall policy exploits this information by recalling more from episodic memory when the familiarity signal is high and recalling less from episodic memory when the familiarity signal is low. This result provides a normative account of experimental findings showing that familiar stimuli shift the hippocampus into a "retrieval mode" where it is more likely to (subsequently) retrieve episodic memories (Duncan et al., 2012; Duncan et al., 2019; Duncan & Shohamy, 2016; Hasselmo & Wyble, 1997; Patil & Duncan, 2018).

In *Simulation 5*, we provide a normative account of recent fMRI findings that suggest that episodic encoding occurs preferentially at event boundaries (Baldassano et al., 2017; Ben-Yakov et al., 2013; Ben-Yakov & Henson, 2018; Reagh et al., 2020). To understand why this encoding policy (of selectively taking "snapshots" at event boundaries) might be useful, we compare this to an alternative encoding policy where the model stores episodic memories both at event boundaries and also in the middle of events. We show that selectively encoding at event boundaries yields better prediction performance, specifically because it leads to a reduced probability of false recall. The intuition for why this occurs is that waiting until the end of the event to encode leads to more information-rich ("complete") snapshots, which are less confusable with other memories. This result leads to a novel experiment proposal, described in *Empirical Prediction 3*.

In *Simulation 6*, we model the intersubject correlation (ISC) results reported in J. Chen et al. (2016): They found that, during the second part of the movie, the ISC between DM participants and RM participants (RM-DM ISC) started out below the RM-RM ISC but it eventually rose to the level of the RM-RM ISC (J. Chen et al., 2016). Moreover, in DM participants, higher levels of hippocampal-cortical interaction at the start of the second part of the movie were associated with higher levels of RM-DM ISC (J. Chen et al., 2016) later in the

second part of the movie. We show that we are able to capture these results by training multiple neural networks, aligning them using a shared response model (P.-H. Chen et al., 2015; Lu et al., 2018), and then performing group-level ISC analysis across the aligned networks. This simulation supports our view that participants in the DM condition recall episodic memories to fill in "gaps" in their situation model, which (in turn) has the effect of bringing their understanding of the story into alignment with the participants in the RM condition (as evidenced by DM-RM ISC rising to the level of RM-RM ISC).

In *Simulation 7*, we simulate the effects of hippocampal damage on event prediction in the model. We find that, if the model is allowed to continuously maintain information in working memory (i.e., there are no distractions/interruptions that clear out the working memory state), the model is sensitive to stimulus history, even if the hippocampus is completely inactivated. This result fits with the results of a recent patient study (Zuo et al., 2020) that showed that an amnesic patient's brain response was sensitive to "scrambling" which parts of a movie preceded the current part, as well as other, more anecdotal results showing that hippocampal patients can comprehend narratives so long as they are not interrupted (Milner et al., 1968).

In *Simulation 8*, we show that the model is capable of integrating information over multiple event sequences to form a more complete situation model via episodic encoding and retrieval. In a recent study, participants listened to a narrative with two storylines that alternated repeatedly (Chang et al., 2020). This study found that the two storylines activated distinct neural patterns that rose and fell at moments when the storyline switched in the narrative (e.g., when the narrative switched from storyline A to B, the "storyline A" pattern faded and the "storyline B" pattern rose). Furthermore, Chang et al. (2020) found that the storyline-specific patterns became more distinct as the narrative progressed. This happens in our model because it stores an episodic snapshot of its working memory state at the end of each segment; when the storyline resumes, the model retrieves episodic memories formed during prior encounters with that storyline, which has the effect of loading previously-encountered storyline features into working memory. If the model then encounters new (previously unseen) storyline features, these are added to the working memory representation alongside the old features, and the old and new features are stored (together) in the next episodic snapshot. In this fashion, the model is able to integrate

new storyline-specific features with old features across segments, making the two storyline representations increasingly distinct from each other.

In *Simulation 9*, we introduce prototypical events (i.e., events that occur more often than other events) in the training environment. Models trained in these environments show classic schema-consistent memory effects: improved memory for the prototypical feature, and also a tendency to falsely remember the prototypical feature in cases where the actual studied feature violated the prototype (Alba & Hasher, 1983). Furthermore, by applying a decoder to the working memory state of the model, we show why it makes these errors: When the prototype is extremely strong, the model shows a tendency to actively intrude the prototype into working memory; the model also shows a tendency to guess the prototypical feature when it is uncertain about the upcoming event – this tendency increases in a graded fashion as a function of prototype strength. We also explore the effect of including these prototypical events on recall policy (i.e., on the setting of the input gate). The model shows a complex pattern of effects modulated by the probability of occurrence of the prototype and the value of the penalty on incorrect predictions; these results form the basis for *Empirical Prediction 4*.

Simulation 1: basic properties of the model

In this simulation, we use the paradigm from J. Chen et al. (2016) (see Figure 5) to illustrate basic points about how the model uses working memory and episodic memory to support event prediction; we also show how the model's uncertainty about upcoming events and the penalty on incorrect predictions influence recall policy.

1.0 Decoding the internal states of the network

To illustrate how working memory and episodic memory are used at the single trial level, we used multivariate pattern analysis (MVPA) to decode the working memory states of the network over time. Figure 8 shows two example trials from the distant memory (DM) condition. Each heatmap shows, for a specific situation feature (e.g., weather), the probabilities assigned to each of that feature's values over time. Namely, the t-th column of the heatmap shows the classifier-produced probability distribution over different beliefs at time t. The 0-th row

corresponds to the state of "don't know". The rest of the rows correspond to different situation feature values. The green box shows the feature value that was presented to the model. The row of the observation (green box) indicates the observed value of this feature. The column of the observation (green box) indicates the time point when this value was observed. The yellow box shows the model's prediction when it was queried about the upcoming event – so, if the model's prediction is correct, the yellow box (the prediction) will be in the same row as the green box (the observation).

In the first example (Figure 8A), during part one, the model initially does not know the value of this feature (because it has not yet been observed). Once the feature is observed to have value 3, this feature becomes decodable, and it stays decodable (indicating that it is being maintained in working memory). Later on, the model responds correctly when this feature is queried. During part two, this feature is queried before it is observed. However, because of episodic memory, the model is able to retrieve the correct feature value much earlier in the sequence (at time 1) and hold it in working memory, thereby making it possible to respond correctly to the query (yellow box) about the upcoming event.

The second example (Figure 8B) shows that episodic recall can sometimes happen too late to be useful. During part two, the model is queried at time 3, but the feature value has not been recalled yet, so the model makes an incorrect guess (yellow box).

Taken together, these two examples show that 1) the model can maintain observed features using its working memory; and 2) the model can use episodic memory to retrieve feature values of the ongoing situation before they are observed.

1.1 Using episodic memory to guide event prediction

The behavioral results for the three task conditions are shown in Figure 9. In the RM condition, the model's prediction accuracy is at ceiling throughout part two of the sequence; the model performs at ceiling because all relevant features of the ongoing situation were shown during part one of the sequence, and the model is able to maintain these features in working memory. In the DM condition, prediction accuracy starts low and jumps to the ceiling after a few time steps. In the NM condition, the model's prediction accuracy increases linearly, reflecting the fact that

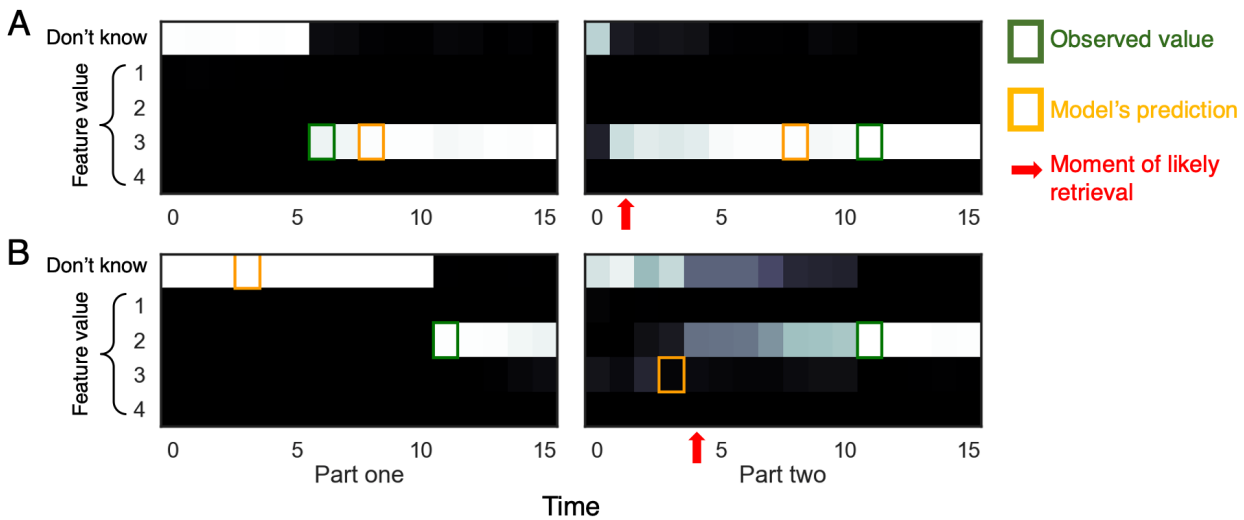

Figure 8

Example – decoded internal states over time. Each heatmap shows the the decoded value of a particular feature over time; each column within the heatmap shows the probability distribution over decoded feature values at a particular time point (the brighter the color, the higher the probability). Parts A and B depict two example trials. A) During part one, the model maintains the feature value in working memory after it is observed, all the way to the end of part one (when the feature value is stored in episodic memory). During part two, the model retrieves the correct feature value from episodic memory, so the feature value becomes decodable before the model observes that feature. This retrieved feature value helps the model to correctly predict the upcoming event. B) Though the model encodes the correct feature value during part one, recall occurs too late to be useful during part two. Because the feature is queried before the feature value is recalled, the model makes an incorrect guess. Moments of likely retrieval from episodic memory are highlighted with red arrows on the two plots.

the model observes more situation features as the event sequence unfolds. The model performs better in the DM condition than the NM condition because the model is able to use episodic memory to retrieve features of the ongoing situation before they are observed, thereby boosting event prediction.

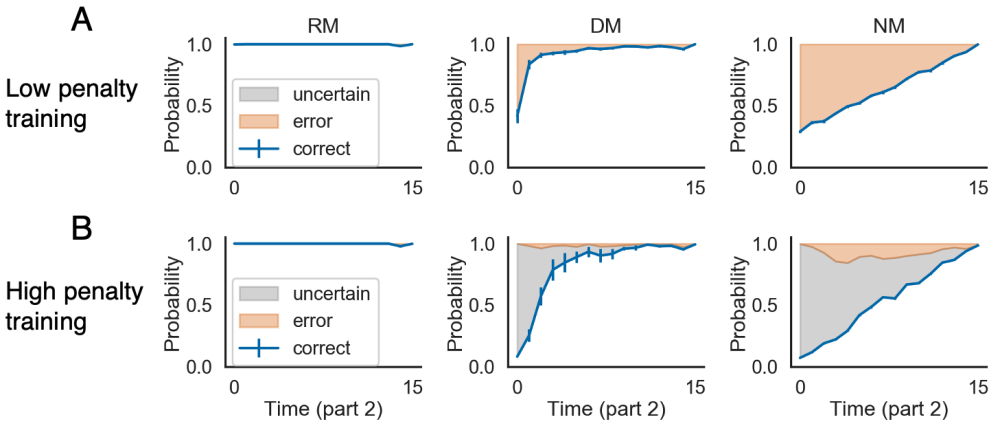

Figure 9

Prediction performance during part 2 for the three conditions, as a function of penalty. Event prediction performance during the second part of the sequence for models trained in A) low penalty and B) high penalty environments. In the recent memory (RM) condition, event prediction accuracy is at ceiling starting from the beginning of part 2 – all situation feature values for the ongoing situation were observed during the first part of the sequence, and the model is able to hold on to these features in working memory. In the distant memory (DM) condition, prediction accuracy starts out much lower, but after a few time steps the accuracy is almost at ceiling. In the no memory condition (NM), prediction accuracy increases linearly, reflecting the fact that the model is able to observe more situation features as the event sequence unfolds. The fact that prediction accuracy is better in the DM condition than in the NM condition suggests that the model is using episodic memory to support event prediction in the DM condition. Regarding penalty effects: In the low penalty environment, the model makes a lot of mistakes. In the high penalty environment, the model uses the "don't know" response (labeled in the figure as "uncertain", shown in gray) to avoid errors. This shows that penalty at training can influence the learned policy. Results are averaged across 15 models; errorbar indicates SEM.

Figure 10 shows episodic memory activation over time during part two, separated by

target and lure. The activation of a given episodic memory is the output of the LCA process, which depends on 1) the cosine similarity value of that memory versus the current hidden state, 2) the level of the input gate, and 3) the level of competition across memories. The target memory is the episodic memory encoded at the end of part one, which was generated by the same situation as part two. Hence, recalling the target memory will support event prediction. The lure memory is the episodic memory encoded at the end of the distractor (situation A) event sequence (see Figure 5).

Across the three conditions, we found that the level of memory activation is much higher in the DM condition than the other two conditions (Figure 10B). Importantly, the model shows more recall in the DM condition than the RM condition, which qualitatively captures the finding that the putative fMRI signature of episodic recall (hippocampal-cortical coupling) was stronger in the DM condition than the RM condition (J. Chen et al., 2016). Note that, in our simulation, the set of available episodic memories in the RM and the DM condition is the same. The main difference is that, in the RM condition, the network has a fully-specified situation model actively maintained in its working memory (the recurrent activity of the LSTM) during part two, which is sufficient for the network to predict upcoming state. In contrast, at the beginning of the DM condition, the network's ongoing situation model is empty – the values for all features are unknown. Overall, this result shows that the learned recall policy is "demand sensitive". That is, the model recalls when there are gaps in its situation model, which cause it to be uncertain about what will happen next. Conversely, the model is less likely to recall when the relevant feature values of the situation model are maintained in working memory, and thus it is more certain about what will happen next.

This policy can be explained in terms of a simple cost-benefit analysis: When the model is unsure about what will happen next, the potential benefits of episodic recall are high. In the absence of recall, the model will have to guess or say "don't know", but if it consults episodic memory, the model could end up recalling the feature of the situation that controls the upcoming event transition, allowing it to make a correct prediction. By contrast, when the feature of the situation that controls the transition is already in working memory (and consequently the model is able to make a specific prediction about what will happen next), there is less of a benefit

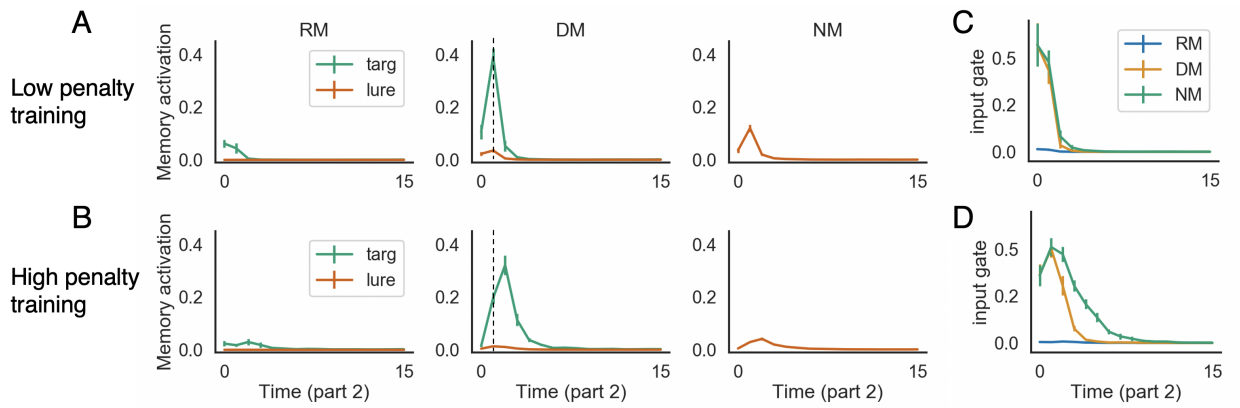

Figure 10

Memory activation and input gate values for the three conditions, as a function of penalty. We compared the activation of target memories versus lure memories during part two of the sequence for models trained in A) low penalty and B) high penalty environments. Target memories were previously encoded in the same situation as the ongoing event sequence. Lure memories were encoded in a different situation. The activation of a given memory is the LCA output for that memory, which depends on the cosine similarity between this memory and the current cell state, the level of input gate, and the level of competition from other memories. The result shows that the overall level of memory activation is stronger in the DM condition than the RM and the NM conditions, which captures the result from J. Chen et al. (2016) that hippocampal-cortical connectivity was stronger in the DM condition than the other two conditions. The finding that recall is suppressed in the RM condition (despite the availability of a relevant episodic memory) can be explained in terms of the model learning to keep the input gate closed in the RM condition (see parts C and D for input gate values). Regarding penalty effects: In the high penalty environment, the average recall time is delayed in the DM condition, allowing the model to accumulate more information before recalling (the dashed line marks the peak of the memory activation curve in the low penalty condition). This delay in recall is attributable to the model learning to open the input gate later in the high penalty condition (compare parts C and D). Results are averaged across 15 models; errorbar indicates SEM.

associated with recall – the only way that recall will help is if the model is holding the wrong feature in working memory and the episodic memory overwrites it. Furthermore, in this scenario, there is also a potential cost to recalling: Lures are always present, and if the model recalls a lure this can overwrite the correct information in working memory. Since the potential costs of recalling outweigh the benefits of recalling in the "high certainty" scenario, the model learns a policy of waiting to recall until it is uncertain about what will happen next.

1.2 The learned policy adapts to the average penalty associated with misprediction during training

A key question is how the learned policy for prediction and recall changes to adapt to different environmental regimes. In this simulation, we varied the level of penalty (i.e., the cost of making incorrect predictions). In the low penalty condition, the penalty value was zero for both training and testing. In the high penalty condition, the penalty value was always four for both training and testing. We expected the learned policy in the high penalty condition to be more conservative.

We found that, when the penalty is low, the model makes specific event predictions (i.e., it refrains from using the "don't know" response) even when it can not reliably predict the next event, leading to many errors (Figure 9A). In contrast, when the penalty is high, the model makes more "don't know" responses (Figure 9B). This strategy is rational – when the penalty is low, the expected reward is larger for randomly guessing an answer than for saying "don't know", but when the penalty is high, the expected reward is larger for saying "don't know" than for random guessing.

The level of the penalty also affects when recall happens. We measured the average memory activation for models trained in low and high penalty environments (Figure 10). The results show that model's average recall timing is delayed when the penalty is high (Figure 10B vs. Figure 10A). The delayed recall time is caused by a corresponding delay in opening the input gate for episodic memories (Figure 10D vs. Figure 10C). To quantify the difference in recall time across the two penalty conditions, we operationalized recall time as the center of mass of the input gate values over time. The recall time delay (recall time in high penalty minus low penalty)

was 0.781 (95%CI 0.51, 1.21); the delay was statistically significant according to a paired-sample t test ($t(14) = 4.41$, $p = 0.0007$).

This delay in recall can be explained in terms of a speed-accuracy trade-off. Waiting longer to recall allows the model to observe more features, which helps to disambiguate the present situation from other, related situations and thereby reduces false recall. However, waiting longer to recall also carries an opportunity cost – the model has to forego all of the rewards it would have received (from correct prediction) if it had recalled earlier. When the penalty is low, the benefits of recalling early (in terms of increased correct prediction) outweigh the costs (in terms of increased incorrect prediction due to false recall), but when the penalty is high, the costs outweigh the benefits, so the model is more cautious and it waits to observe more features to be sure that the memory it (eventually) recalls is the right one.

1.3 Interactions between penalty and uncertainty

Having demonstrated (above) that both uncertainty and penalty affect the model's learned recall policy, we then explored how these factors interact. To accomplish this, we took the activation values for target memories in the DM condition, and we separated the data based on whether or not the queried situation feature had already been observed (during part two) – our logic here was that the model would be more uncertain (and thus more likely to call upon episodic memory to "fill the gap") if the feature had not already been observed, compared to if the feature had been observed; and we crossed this manipulation of uncertainty with our penalty manipulation.

The results of the simulation are shown in Figure 11. In keeping with the results reviewed above, levels of memory activation are higher overall when the queried feature has not already been observed during part two ("high uncertainty"), compared to when the queried feature has been recently observed; likewise, recall is delayed in the high penalty condition compared to the low penalty condition (Figure 11A vs. B). These patterns of target memory activation are driven by corresponding modulations of the input gate (Figure 11C vs. D). Importantly, there is also an interaction between uncertainty and penalty, whereby recall is modulated more strongly by uncertainty in the high penalty condition than in the low penalty condition. As discussed above,

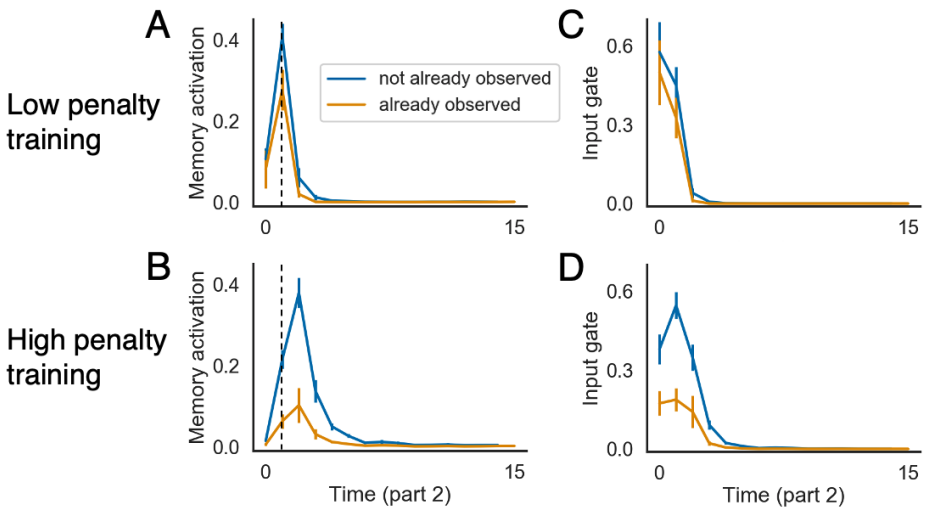

Figure 11

The effects of uncertainty and penalty on recall policy interact. Parts A and B show activation for target memories in the distant memory (DM) condition, separated according to whether the relevant situation feature has already been observed (or not) during part 2 of the sequence. The result is separately plotted for models trained in A) low penalty and B) high penalty environments. Recall levels are higher overall when the queried information has not been observed (i.e., when there is high uncertainty) and recall is delayed overall in the high penalty condition (the dashed line marks the peak of the memory activation curve in the low penalty condition). Furthermore, there is an interaction whereby the effect of uncertainty on memory activation is higher in the high penalty condition than in the low penalty condition. These patterns of target memory activation are driven by corresponding modulations of the input gate, shown in parts C and D. Results are averaged across 15 models; errorbar indicates SEM.

the model avoids recall when it is certain about the upcoming event because of the risk that it will recall a lure memory that overwrites the correct feature in working memory, causing an error. Reducing the penalty reduces the potential cost of this "overwriting" and thus the model is more willing to risk overwriting (by recalling-when-already-certain) in the low penalty condition, compared to the high penalty condition.

Empirical prediction 1 – how episodic retrieval is modulated by uncertainty

Future experimental work could test these predictions about how uncertainty modulates recall by presenting participants with artificially-generated narratives that mirror the stimuli used in these simulations. Concretely, consider the example in Figure 4. Participants could listen to stories where transitions between events are governed by particular features (e.g., the barista at the coffee shop is more likely to insult your drink order if the weather is rainy vs. sunny). After learning these contingencies (based on exposure to a large number of stories), participants could be asked to predict the barista's response either when they already know the feature that determines the barista's mood (i.e., whether it is rainy or sunny) or when they do not. Based on our modeling results, we predict that participants will be more likely to consult episodic memory (as indexed by increased hippocampal-cortical connectivity, similar to the method in J. Chen et al., 2016) when they do not know the mood-determining feature (the weather) than when they do know the feature.

Simulation 2: The learned policy is sensitive to the current penalty level during test

In the previous simulation, the distribution of penalty values at training was confounded with the distribution of penalty values at test; that is, a model was either a) both trained and tested in a "high penalty" environment (where penalty was fixed at 4), or b) both trained and tested in a "low penalty" environment (where penalty was fixed at 0). Because of this confound, it is unclear how much the model's policy was determined by the penalties it was exposed to at training vs. the penalties it was exposed to at test. To unconfound these factors, we ran a simulation where all models were trained in the same environment, in which they were exposed to a range of penalty values uniformly sampled between 0 and 4; we then manipulated the penalty

value during part two of the sequence at test. Importantly, during part one of the sequence, the penalty value was always set to the average value (2). We did this to ensure that the model could not leverage its knowledge of the part-two penalty when deciding how to encode part one.

One possibility is that a model trained in this scenario would learn an inflexible policy that tracks the average penalty at training and is insensitive to the level of penalty during part two of the sequence at test. Another possibility is that a model exposed to a range of penalties at training can effectively learn a "library" of policies that are appropriate for different penalty scenarios, and then flexibly deploy them at test based on the current penalty.

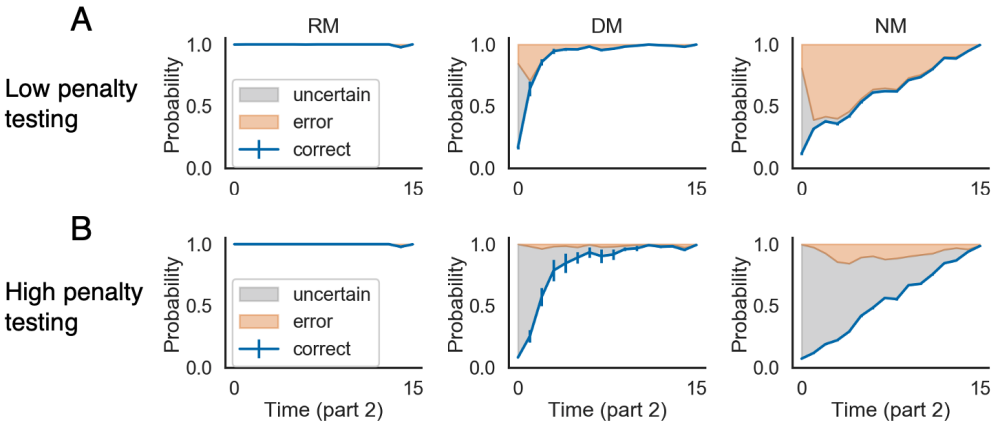

Figure 12

The level of penalty at test influences the current recall policy. Event prediction performance when setting the current penalty to be A) low vs. B) high. The result is largely analogous to Figure 9. In the low penalty environment, the model makes a lot of mistakes. In the high penalty environment, the model uses "don't know" responses to avoid errors. Results are averaged across 15 models; errorbar indicates SEM.

The results of this simulation are shown in Figure 12. When the penalty level at test is high (versus low), the model shows a lower error rate and makes more "don't know" responses – whenever the model makes a specific prediction, it tends to be correct. Finally, as in *Simulation 1*, to quantify the difference in recall time across the two penalty conditions, we operationalized recall time as the center of mass of the input gate values over time. The recall time delay (recall time in high penalty minus low penalty) was 0.453 (95%CI 0.3, 0.653); the delay was statistically significant according to a paired-sample t test ($t(14) = 4.94$, $p = 0.0003$). Taken together, these

results demonstrate that the model is capable of flexibly adjusting its policy at test based on the current penalty.

Simulation 3: Recall is sensitive to the interaction between event similarity and penalty

In this simulation, we show how the similarity of event memories affects recall policy. To manipulate similarity, we varied the proportion of shared situation feature values across event sequences during training. In the low-similarity condition, the similarity between the distractor situation (i.e., situation A; see Figure 5) and the target situation was constrained to be less than 40%, so target memories and lures were relatively easy to distinguish. In the high-similarity condition, the similarity between the distractor situation and the target situation was constrained to fall between 35% and 90%. We used a rejection sampling approach to implement these similarity bounds – during stimulus generation, we kept generating distractor event sequences until they fell within the similarity bounds with respect to the target sequence. Otherwise, the simulation parameters were the same as in *Simulation 2*.

In the high-similarity condition, target and lure memories were more confusable, and thus the risk of lure recall was higher. In light of this, we expected that the model would adopt a more conservative recall policy (i.e., recalling less) in the high-similarity condition. We also expected that this effect would be stronger when the penalty is high; when the penalty is low, there is less of a cost for recalling the lure memory, and thus less of a reason to refrain from recalling in the high-similarity condition.

We compared the recall policies in a 2 x 2 penalty (high vs. low) by similarity design (high vs. low). For the penalty manipulation, each model was trained on a range of penalty values from zero to four, then tested on low (zero) vs. high (four) penalty, as in *Simulation 2*. Figure 13 shows the memory activations over time in the DM case for all four conditions. As expected, levels of recall are lower in the high-similarity condition, especially when the penalty is high. Penalty has no discernible effect on recall in the low-similarity condition. Note that the lack of penalty effects in the low-similarity condition differs from what we observed in *Simulations 1 and 2*, where recall was delayed in the high-penalty (vs. low-penalty) condition. The only

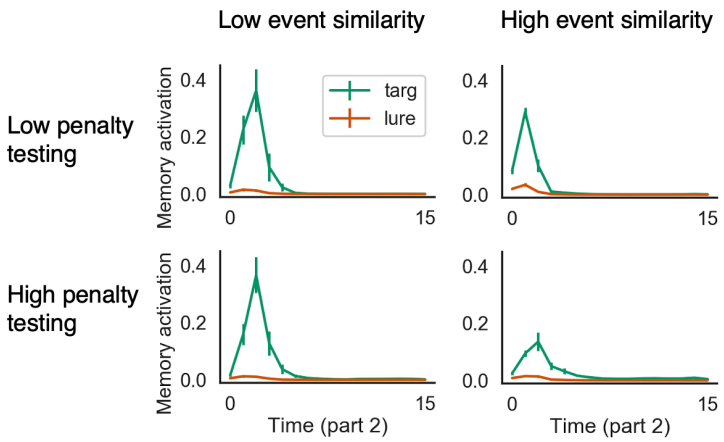

Figure 13

The effect of event similarity and penalty on recall policy. Memory activation over time in the DM condition, for models trained in four different environmental regimes, defined by the crossing of penalty (low vs. high) x event similarity (low vs. high). The effect of penalty depends on event similarity (which controls the confusability of episodic memories). When event similarity is high (leading to a higher risk of recalling lures) the model shows reduced levels of recall, especially when the penalty is high. On the other hand, when event similarity is low, the model shows equivalently strong levels of recall in the low and high penalty conditions. Results are averaged across 15 models; errorbar indicates SEM.

difference between the low-similarity condition of this simulation and *Simulation 2* is that target-lure similarity was sometimes (by chance) higher than 40% in *Simulation 2*, but it was never higher than 40% here. Effectively, the model here has learned that – if lures are never highly similar to the target – it is "safe" to recall at full strength, even with a high penalty.

Empirical prediction 2 – recall policy is sensitive to memory confusability and the cost of mistakes

To test the model's predictions about how recall policy should be affected by memory confusability and penalty, we can manipulate 1) the penalty for making incorrect predictions, by manipulating monetary rewards and 2) the similarity across event memories, by manipulating the proportion of shared features across events. We can use synthetic stories as the stimuli (as in *Empirical Prediction 1*), allowing us to precisely control the amount of feature overlap across

stories. To measure when recall occurs, we can measure hippocampal-cortical inter-subject functional connectivity, as in J. Chen et al. (2016). As discussed in *Simulation 3*, our model predicts that levels of recall will depend on the interaction between the similarity across memories and penalty for making incorrect predictions: When similarity is reliably low, levels of recall should be high regardless of the penalty; when similarity is reliably high, levels of recall should be lower in the high-penalty condition than the low-penalty condition.

Simulation 4: A familiarity signal can guide episodic recall policy

Prior work has demonstrated that cortex is capable of computing a familiarity signal on its own (i.e., without hippocampus) that discriminates between previously encountered and novel stimuli (Holdstock et al., 2002; Norman, 2010; Norman & O'Reilly, 2003; Yonelinas, 2002). In this section, we study how this cortical familiarity signal can support episodic recall policy. Relevant to this point, several recent studies have found that encountering a familiar stimulus can temporarily shift the hippocampus into a "retrieval mode" where it is more likely to retrieve episodic memories in response to available retrieval cues (Duncan et al., 2012; Duncan et al., 2019; Duncan & Shohamy, 2016; Hasselmo & Wyble, 1997; Patil & Duncan, 2018). Here, we assess whether our model can provide a normative account of these "retrieval mode" findings.

Intuitively, familiarity can guide episodic recall policy by providing an indication of whether a relevant episodic memory is available. If an item is unfamiliar, this signals that it is unlikely that relevant episodic memories exist, hence the expected benefit of recalling from episodic memory is low (if there are no relevant episodic memories, episodic recall can only yield irrelevant memories, which lead to incorrect predictions); and if an item is familiar, this signals that relevant episodic memories are likely to exist and hence the benefits of recalling from episodic memory are higher. These points suggest that the model would benefit from a policy whereby it adopts a more liberal criterion for consulting episodic memory when stimuli are familiar as opposed to novel.

To test this, we ran simulations where we presented a "ground truth" familiarity signal to the model during part two of the sequence. The familiarity signal was presented using an additional, dedicated input unit (akin to how we present penalty information to the model).

Specifically, during part two, if the ongoing situation had been observed before (as was the case in the RM and DM conditions), the familiarity signal was set to one. In contrast, if the ongoing situation was novel (as was the case in the NM condition), then the familiarity signal was set to negative one. Before part two, the familiarity signal was set to zero (an uninformative value). Other than these changes, the parameters of this simulation were the same as *Simulation 2*. The model was tested on penalty value of 2 – the average of the training range.

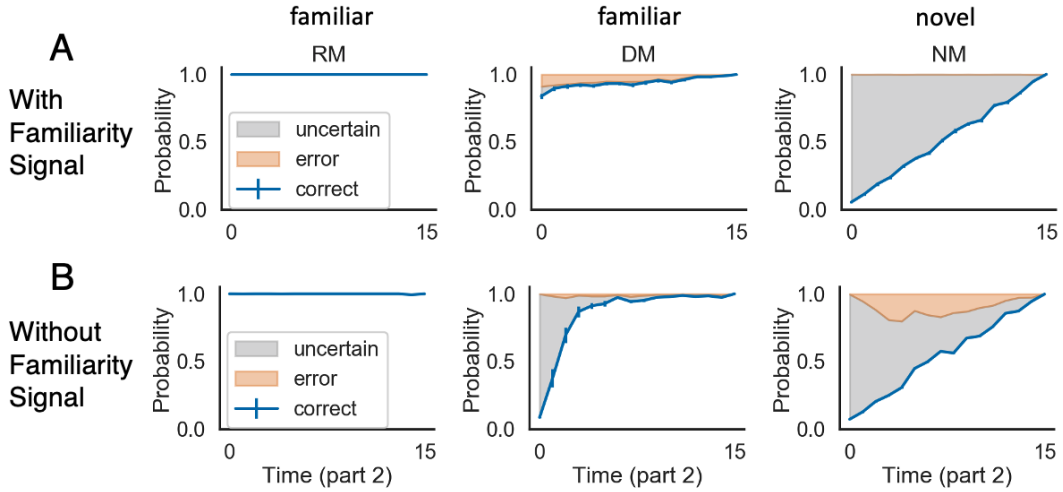

Figure 14

The familiarity signal can improve event prediction. Event prediction performance for models with (A) vs. without (B) access to the familiarity signal. With the familiarity signal (A), the model shows 1) higher levels of correct prediction in the DM condition, and 2) a reduced error rate in the NM condition, compared to the model without the familiarity signal. Results are averaged across 15 models; errorbar indicates SEM.

Figures 14 and 15 illustrate prediction performance, memory activation, and input gate values for models with and without the familiarity signal. When the model has access to a veridical familiarity signal (+1 for RM and DM, -1 for NM), it opens the input gate immediately and strongly in the DM condition (Figure 15D - DM), leading to higher activation of both the target memory and the lure (Figure 15A - DM) in the DM condition, relative to models without the familiarity signal (Figure 15B - DM). Behaviorally, models with the familiarity signal show both a higher correct prediction rate and a slightly higher error rate in the DM condition, compared to models without the familiarity signal (Figure 14A vs. B - DM). This slight increase

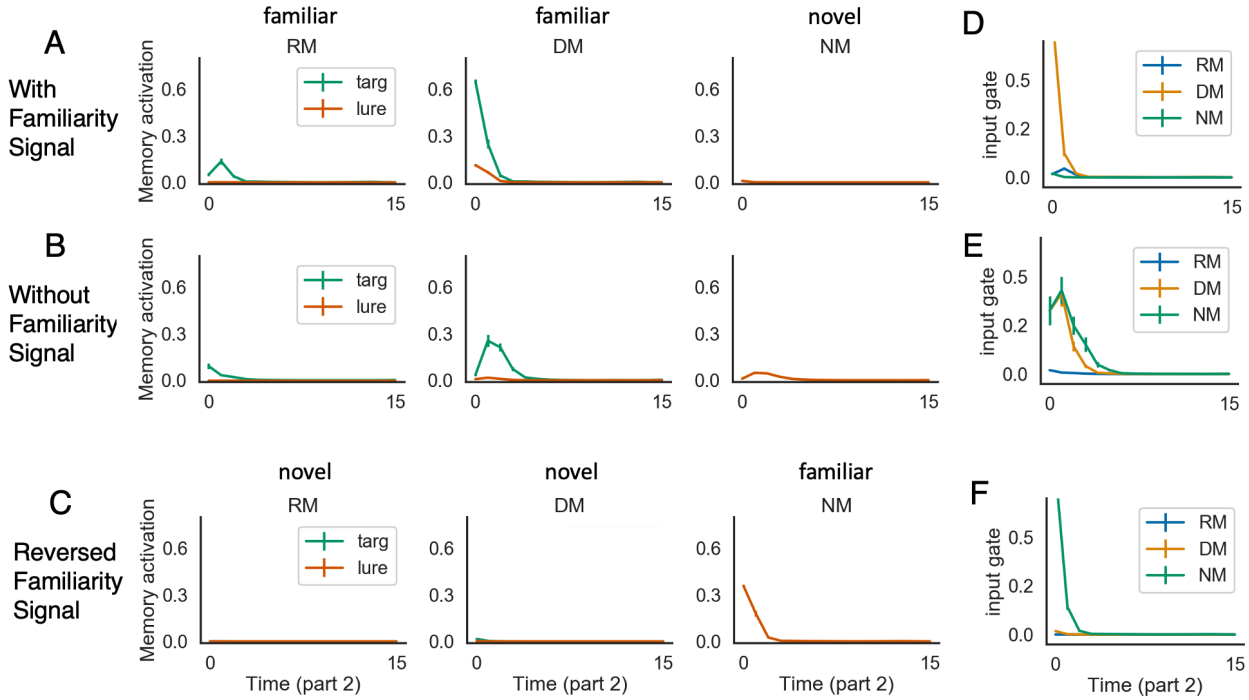

Figure 15

Recall is modulated by familiarity. This figure shows the memory activation and input gate values over time for three conditions: 1) with the familiarity signal (A, D), 2) without the familiarity signal (B, E), and 3) with a reversed (opposite) familiarity signal at test (C, F). With the familiarity signal (A), the model shows higher levels of recall in the DM condition, and suppresses recall even further in the NM condition, compared to the model without the familiarity signal (B). This is due to the influence of the input gate – the model with the familiarity signal recalls immediately in the DM condition, and turns off recall almost completely in the NM condition (D). Note also that levels of recall in the RM condition stay low, even with the familiarity signal (see text for discussion). Finally, parts C and F show that reversing the familiarity signal at test suppresses recall in the DM condition and boosts recall in the NM condition. Results are averaged across 15 models; errorbar indicates SEM.

in errors occurs because, when the model recalls immediately during part two, the model (in some cases) has not yet made enough observations to distinguish the target and the lure. In the NM condition, with the familiarity signal, the model keeps the input gate almost completely shut (Figure 15D - NM). Consequently, the level of memory activation stays very low in the NM condition (Figure 15A - NM), which reduces the error rate in the NM condition to zero (Figure 14A - NM). The RM condition is an interesting case: Previously (in *Simulation 1*), we found that the model refrained from recalling in the RM condition; we found that the same pattern is present here, even when we make a familiarity signal available to the model: Input gate and memory activation levels are both very low (Figure 15A, D - RM), similar to models without access to the familiarity signal (Figure 15B, E - RM). This shows that model does not *always* recall when given a high familiarity signal – in this case, the presence of relevant information in working memory (which suppresses recall) "overrides" the presence of the familiarity signal (which enhances recall in the DM condition).

Finally, we can trick the model into reversing its recall policy by reversing the familiarity signal at test (Figure 15C, F). In this condition, the (reversed) signal indicates that the ongoing situation is novel (-1) in the RM and the DM condition, and the ongoing situation is familiar (+1) in the NM condition. As a result, the model suppresses recall in the RM and DM conditions, and recalls lures in the NM condition.

Overall, the results of this simulation show that our model is able to use a familiarity signal to inform its recall policy in the service of event prediction. Consistent with empirical results (Duncan et al., 2012; Duncan et al., 2019; Duncan & Shohamy, 2016; Hasselmo & Wyble, 1997; Patil & Duncan, 2018), we found that the model recalls more when the ongoing situation is familiar, unless the model has low uncertainty about the upcoming event. These results provide a normative account of how the brain can exploit a familiarity signal to enhance its policy for episodic recall.

Simulation 5: Encoding selectively at event boundaries benefits subsequent recall

The previous simulations focused on how various factors affect the model's learned retrieval policy; here, we shift our focus to the complementary question of *encoding policy*: When

is the best time to store episodic memories? As discussed in the *Introduction*, several recent fMRI studies have found that hippocampal response peaks tend to align with subjectively annotated event boundaries, produced by a separate set of subjects (Baldassano et al., 2017; Ben-Yakov and Henson, 2018; Figure 16A). This boundary-locked response is believed to be a neural signature of episodic encoding, since its magnitude predicts subsequent memory performance (Baldassano et al., 2017; Ben-Yakov and Dudai, 2011; Reagh et al., 2020; Figure 16B, C, D). Taken together, these findings suggest that encoding takes place selectively at event boundaries (for converging evidence from ECoG, see Michelmann et al., 2020). Based on these results, we opted in our previous simulations to have the model take a single episodic snapshot at the end of part one of the sequence (rather than taking snapshots throughout the sequence).

In this simulation, we critically assess the assumption that encoding occurs selectively at event boundaries, by comparing models incorporating this assumption to alternative models that also encode an extra episodic memory midway through the first sequence. If selectively encoding at event boundaries yields better performance, this would provide a normative justification for the empirical findings reviewed above. All other simulation parameters are the same as in *Simulation 2*.

The results of the simulation (Figure 17) show that, indeed, models that store a single episodic memory at the end of part one perform better at event prediction in the DM condition, compared to models that also store an extra episodic memory midway through part one (Figure 17). In the RM and NM conditions (which, as discussed in *Simulation 1*, do not strongly engage episodic memory), performance does not differ across the two encoding regimes. These results raise the key question of why performance is worse in the DM condition when the model encodes an extra snapshot midway through the first event; naively, one might think that more snapshots would yield better memory. The key to understanding this finding is that – when an event starts out – it may not be clear what is happening; the longer you wait, the more information is revealed. Consequently, if the model waits to take the snapshot, the snapshot will contain more information about what is going on, and it will therefore be less confusable with other memories. Conversely, if the model takes the snapshot at an earlier point in time, the snapshot will contain less information and therefore will be more confusable with other memories, leading to recall of

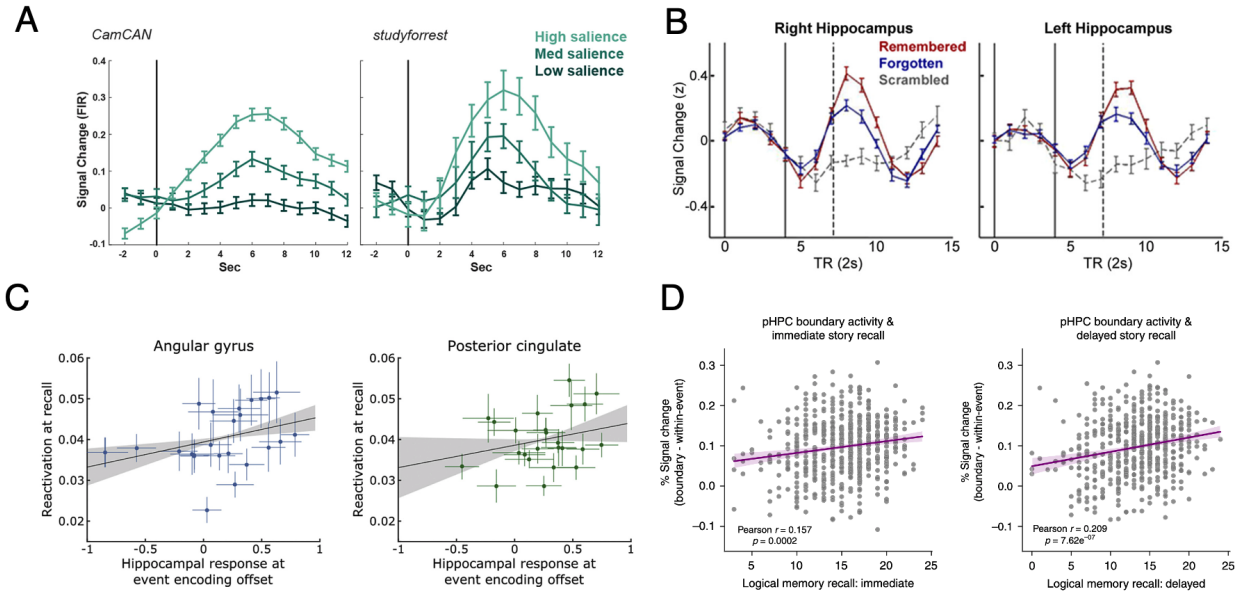

Figure 16

Empirically, the neural signature of episodic encoding is time-locked with event boundaries. Parts A, B, C, and D reprinted with permission. A) Results from Ben-Yakov and Henson (2018): The magnitude of the hippocampal response was modulated by boundary saliency, which indicates the level of cross-subject agreement that an event boundary occurred. B) Results from Ben-Yakov and Dudai (2011): The magnitude of the hippocampal response at the end of an event was higher for subsequently remembered (vs. forgotten) events. C) Results from Baldassano et al. (2017): The magnitude of the hippocampal response at event boundaries was correlated with the level of pattern reinstatement during subsequent recall. D) Results from Reagh et al. (2020): The magnitude of the hippocampal response at event boundaries was correlated (across participants) with subsequent memory performance.

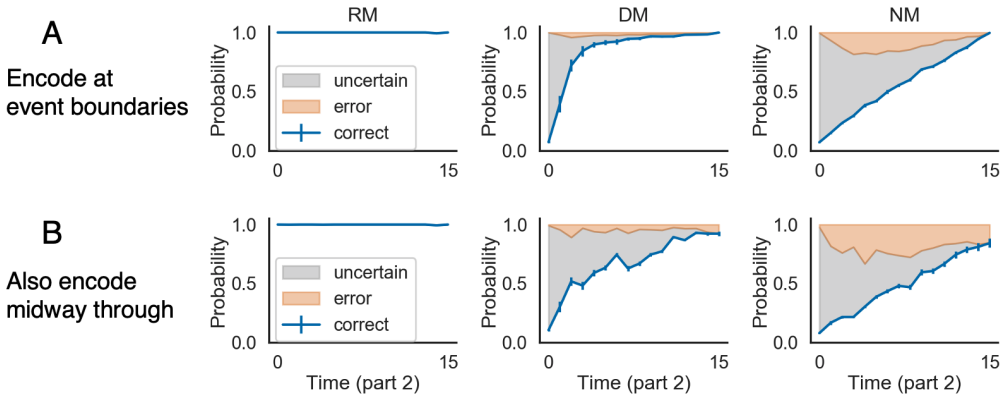

Figure 17

Models that selectively encode at event boundaries perform better. Event prediction performance during part two of the sequence for A) models that selectively encode at event boundaries vs. B) models that also encode midway through part one of the sequence. In the DM condition (where episodic memory is highly useful), event prediction performance is better for models that selectively encode at event boundaries, compared to models that also encode midway through the first event sequence. Results are averaged across 15 models; errorbar indicates SEM.

lures and (through this) prediction errors.

To make this idea concrete, consider the toy example illustrated in Figure 18. In this example, the model first observes an event sequence: It observes that the location is a house, the weather is sunny, and the person is happy (Figure 18A bottom). If the model encodes an episodic memory at the event boundary, this memory will contain all three situation features. In comparison, if the model also encodes midway through the event sequence, the model will have an additional memory that encodes a partial segment of the event sequence, which (in this example) contains the location and weather but does not contain the person's mood. Later on, the model observes a second event sequence (Figure 18B bottom). Since this event sequence came from a different situation (i.e., two of the situation features are different), the model should (ideally) not recall any memory about the first sequence. As the second event sequence unfolds, suppose the model decides to recall after it observed the person's mood and the location. The small memory chunk (Figure 18B top, boxed in red) cannot be rejected, since it contains one matching feature (location = house) and no mismatching features. In contrast, if the model selectively encodes at

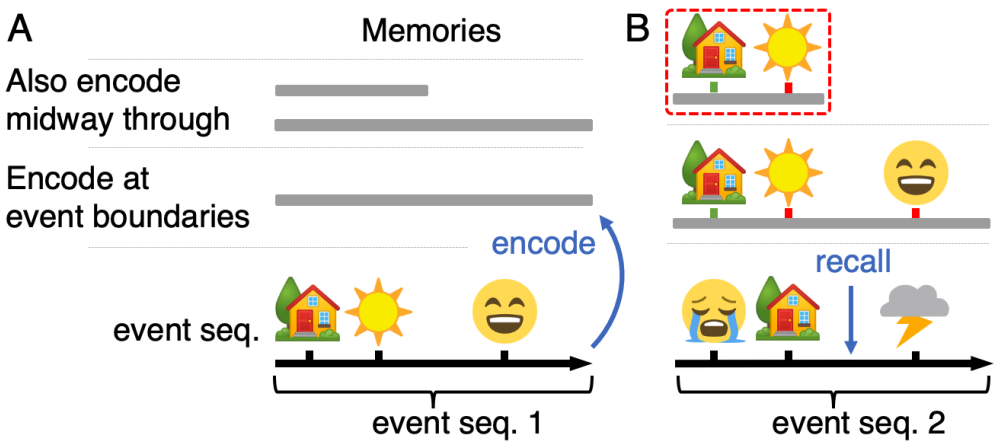

Figure 18

Why encoding selectively at event boundaries might help subsequent recall. A) The resulting memory chunks under the two encoding regimes. Encoding selectively at event boundaries (in blue) leads to an episodic memory that binds together all of the features of the event sequence. If the model also encodes in the middle of an event sequence, the model ends up with an additional partial memory that only records the first part of the event sequence. B) Encoding within an event sequence can lead to subsequent lure recall, whereas encoding a complete situation model makes lures easier to reject. In this example, the character's mood is essential to reject the lure memory. The partial memory (top row), encoded in the middle of the first event sequence, contains the location and weather but does not contain the character's mood, so there is no mismatch with the model's observations so far. This smaller memory chunk cannot be correctly rejected, leaving the model vulnerable to lure recall.

event boundaries, the model will end up with an episodic memory with a complete situation model that contains the person's mood (Figure 18B middle), which allows this memory to be correctly rejected. This example shows how binding all situation features together can help with subsequent lure rejection. Importantly, the model results support this idea that "small memory chunks" (resulting from taking a snapshot midway through the event) lead to false recall: When we examined the level of memory activation, we found that models that encoded midway through an event sequence had a higher level of lure activation (Figure 19).

To summarize the results from this simulation, the model does better when we force it to

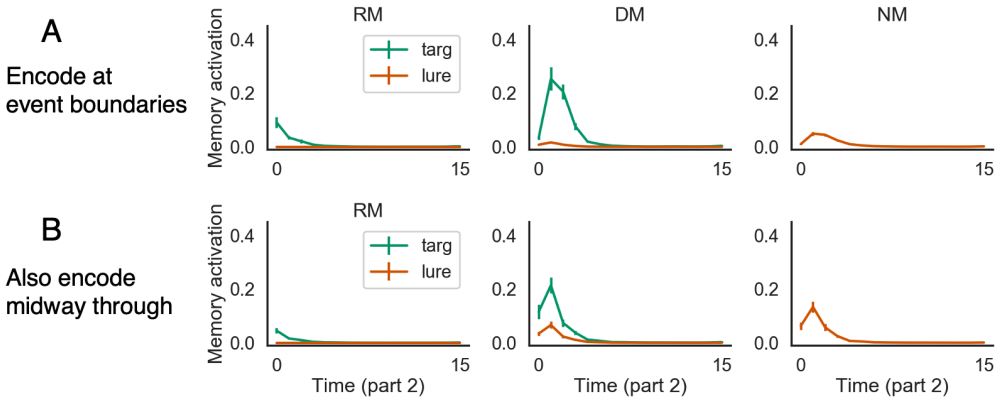

Figure 19

Models that encode in the middle of an event have higher lure activation. Memory activation during part two of the sequence for A) models that selectively encode at event boundaries vs. B) models that also encode midway through the first event sequence. Models that also encode midway through the first event sequence show higher levels of lure activation. Results are averaged across 15 models; errorbar indicates SEM.

wait until the end of an event to take a snapshot; waiting to encode leads to more information-rich ("complete") snapshots that are less confusable with other memories. This model result provides a normative justification for the results from Ben-Yakov and Dudai (2011) and Baldassano et al. (2017) showing preferential encoding-related hippocampal activity at the end of events.

Empirical prediction 3 – the influence of encoding timing on subsequent recall

To test the prediction that memories encoded mid-event are more likely to cause confusion during retrieval, we can introduce artificial event boundaries to trigger additional encoding during movie viewing and examine if those memories cause recall errors. Previous studies have shown that presenting a blank screen between movie clips is sufficient to trigger episodic encoding (Ben-Yakov & Dudai, 2011; Ben-Yakov et al., 2014) – specifically, these studies found that participants' hippocampal activity peaked after the blank screen, and the magnitude of these peaks was larger for subsequently-remembered events than subsequently-forgotten events (Ben-Yakov & Dudai, 2011).

Figure 20 shows the design of our proposed experiment. We can use a blank screen

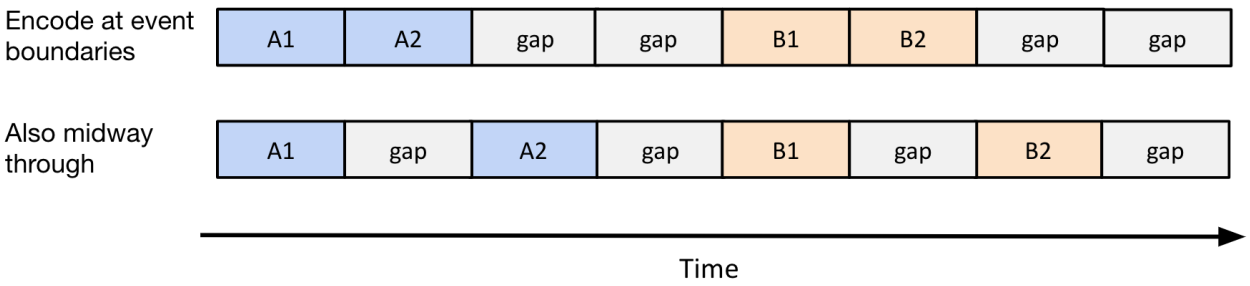

Figure 20

Proposed experiment design to test the influence of encoding timing on subsequent recall. Participants can watch two-part movies (A1 and A2, or B0 and B1) either continuously or interrupted midway through (between the two parts) by a blank screen (gap). We hypothesize that the inserted gap will trigger episodic encoding, resulting in participants forming less complete (more fragmentary) episodic memories; we further hypothesize that these fragmentary memories will be more likely to be recalled in the wrong situation, leading to more prediction errors later (see text for explanation).

("gap") to trigger a memory snapshot mid-event, and we can construct stimuli so the resulting (partial) memory will spuriously match a retrieval cue later on (as in Figure 18). In this situation, we expect that the fragmentary memory will be recalled, resulting in increased memory prediction errors (compared to a situation where the blank screen was not presented mid-event).

Simulation 6: Recall fills the gaps of the ongoing situation model

In this simulation, we return to the "Twilight Zone" study of J. Chen et al. (2016) that we discussed earlier, and show that our model can account for the detailed pattern of results from this study. As noted in the *Introduction*, this study performed a temporal inter-subject correlation (ISC) analysis to compare the neural activity time courses between participants in the DM and RM conditions (for the details of ISC, see *Appendix - methods detail - Inter-subject correlation*). Specifically, J. Chen et al. (2016) found that, at the beginning of movie part two, temporal ISC between DM participants and RM participants (RM-DM ISC) was lower than the temporal ISC within RM participants (RM-RM ISC) (Figure 21A), suggesting differences in their understanding of the ongoing situation; over the course of movie part two, DM-RM ISC gradually

converged with RM-RM ISC (Figure 21A). Moreover, the increase in RM-DM ISC was correlated with the level of hippocampal-cortical coupling that was present at the start of movie part two (Figure 21B), suggesting that episodic recall contributed to the convergence in participants' understanding of the ongoing situation.

One possible interpretation of this finding is that RM-DM ISC reflects the similarity between the situation models maintained in RM participants versus DM participants. At the beginning of part two, for RM participants, information about part one was actively maintained in their working memory, so they started with a complete situation model. In contrast, DM participants started with an "empty" situation model (e.g., the agent, location, time were unknown), and they had to fill in the situation feature values during part two. After observing some events, DM participants recalled episodic memories from part one, which filled the gaps in their situation model (making it more similar to the situation model of RM participants).

To simulate these results, we used the trained neural networks (n = 30) from *Simulation 2*, then assigned half of them to the RM condition and half to the DM condition. Then we functionally aligned these neural networks using the shared response model (P.-H. Chen et al., 2015; Lu et al., 2018) and performed the same temporal ISC analysis presented in J. Chen et al. (2016). As discussed earlier, functionally aligning different neural networks is necessary for comparing activity patterns across networks.

Our simulation results qualitatively capture the findings from J. Chen et al. (2016). During part two, DM-RM ISC starts lower than the RM-RM ISC, but as the event sequence unfolds, they gradually became statistically indistinguishable (Figure 21C). Moreover, in the DM condition, the level of memory activation at time t is correlated with the increment in DM-RM ISC from time t to time $t + 1$ (Figure 21D). As a comparison point, in the RM condition (where the model is not relying on episodic recall to fill in gaps in the situation model; see *Simulation 1*), memory activation does not correlate with the change in (RM-RM) ISC. Collectively, these results establish that episodic recall accelerates the convergence between model activations in the DM and RM conditions.

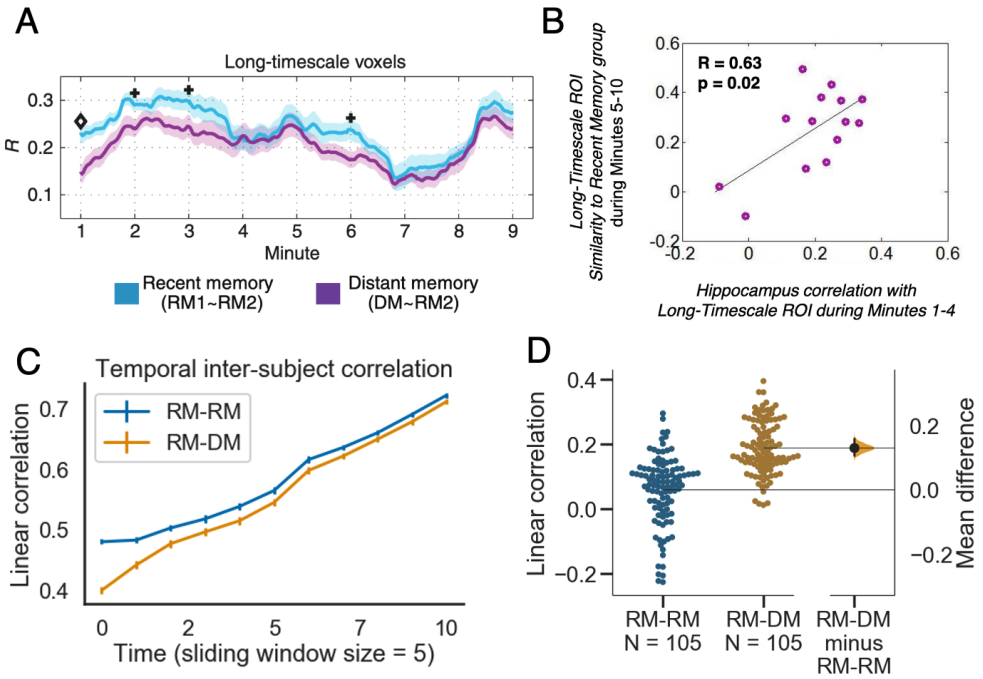

Figure 21

Episodic recall makes the activity dynamics between the DM and RM conditions more similar: model of results from J. Chen et al. (2016). Parts A and B reprinted with permission. A and B show experiment results and C and D show model results. A) The sliding-window temporal inter-subject correlation (ISC) over time, during part 2 of the movie. The recent memory ISC, or RM-RM ISC, was computed as the average ISC value between two non-overlapping subgroups of the RM participants. The distant memory ISC, or RM-DM ISC, was computed as the average ISC between one sub-group of RM participants and the DM participants. Initially, the RM-DM ISC was lower than RM-RM ISC, but as the event sequence unfolded, RM-DM ISC rose to the level of RM-RM ISC. B) For the DM participants, the level of hippocampal-cortical inter-subject functional connectivity at the beginning of part 2 of the movie (minutes 1-4) was correlated with the level of RM-DM ISC later on (minutes 5-10). C) Sliding window temporal ISC in part 2 between the RM models (RM-RM) compared to ISC between the RM and DM models (RM-DM). The convergence between RM-DM ISC and RM-RM ISC shows that activity dynamics in the DM and the RM models become more similar over time (compare to part A of this figure). Results are averaged across 15 models; errorbar indicates SEM.

Figure 21 *(previous page)*

D) The correlation in the model between memory activation at time t and the change in ISC from time t to $t+1$, for the first 10 time points in part 2. In the DM condition, memory activation is correlated with the change in RM-DM ISC (in orange; compare to part B of this figure). In contrast, in the RM condition (where the model does not need to use episodic memory to fill in gaps), the correlation between memory activation and the change in RM-RM ISC is much smaller (in blue). Each point is a subject-subject pair across the two conditions (hence 105 unique pairs). The 95% bootstrap distribution on the side shows that the correlation between memory activation and the change in RM-DM ISC is significantly larger than the correlation between memory activation and the change in RM-RM ISC.

Simulation 7: The cortical network integrates information over long timescales, even without the hippocampus

In *Simulations 1 and 2*, we showed that – in our model – episodic memory is less engaged during continuous processing (i.e., the RM condition), compared to a scenario where there are gaps in the ongoing situation model (i.e., the DM condition); the model appears to be relying almost entirely on working memory in the RM condition. Moreover, during part two, the model's prediction performance in the RM condition is almost perfect. Taken together, these findings suggest that working memory is sufficient to enable almost-perfect performance during part two.

This result is broadly consistent with case studies of amnesic patients with hippocampal damage. It is well known that hippocampal patients are impaired at forming new episodic memories (Corkin, 2002; Gilboa et al., 2006; Milner et al., 1968). In these patients, delays as short as a few seconds are sufficient to lead to performance degradation on tests of memory for simple, abstract stimuli (Holdstock et al., 1995). However, their performance in natural conditions is better than what these experiments would lead one to predict. For example, hippocampal patients can carry out a coherent conversation over several minutes (Milner et al., 1968), which leads to the hypothesis that the cortical network can process continuous natural

stimuli on its own over several minutes (J. Chen et al., 2016).

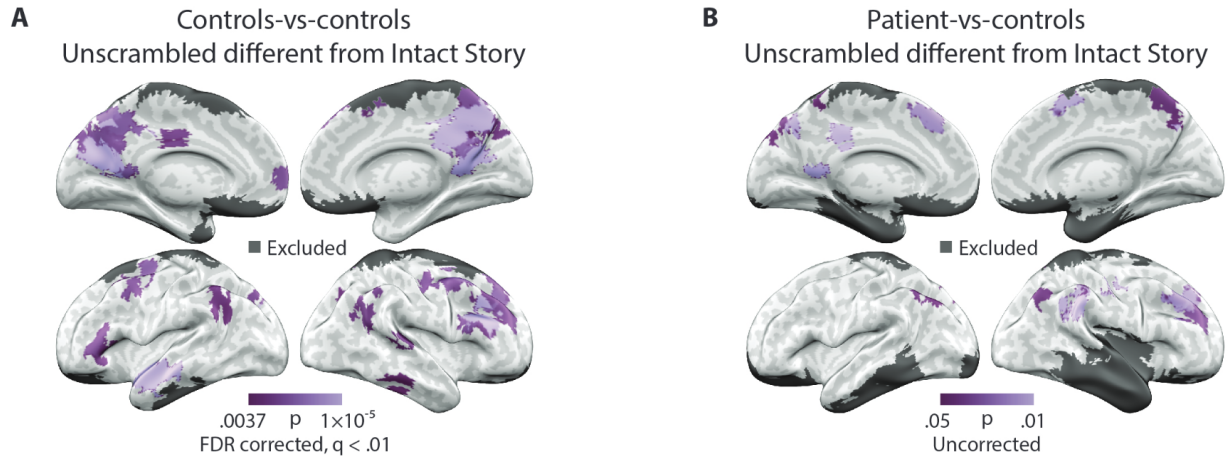

Figure 22

Scrambling sensitivity for controls and patients: empirical results from Zuo et al. (2020). Reprinted with permission. Participants listened to an auditorily-presented narrative both in its original (intact) order and also when it was scrambled at the paragraph level. The fMRI data from the scrambled condition were then "unscrambled" (i.e., put back in the original order) and compared to the fMRI data from the intact condition. The brain maps show the scrambling sensitivity for A) healthy controls and B) a patient with hippocampal damage. Color intensity shows the level of scrambling sensitivity – the extent to which scrambling the stimulus changed the neural activity over time, relative to the neural activity evoked by the intact stimulus. In both the patient and the controls, scrambling the story affected neural activity patterns in DMN regions.

Here, we address the question of how well cortex can retain situation information on its own, by lesioning the episodic memory system in our model and assessing how *temporally scrambling* the sequence of stimuli affects the neural response in the model. Intuitively, if temporal scrambling of a narrative changes the neural response, then the neural response must be history-dependent. Moreover, scrambling the stimuli at different temporal scales makes it possible to measure the length of history-dependency (the *temporal receptive window*; TRW) (Himberger et al., 2018; Lerner et al., 2011). For example, a prior fMRI study scrambled an audiobook at the paragraph-level, sentence-level, and word-level (Lerner et al., 2011). Their results show that DMN

activity is sensitive to paragraph-level scrambling, meaning that the instantaneous DMN activity depends on the information from the previous paragraph (Lerner et al., 2011).

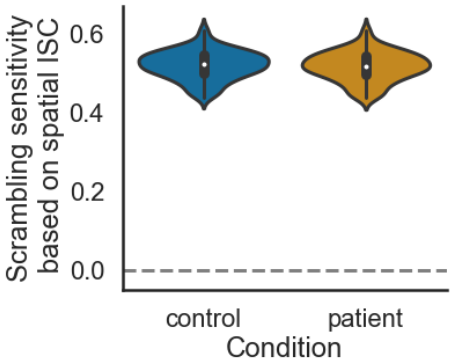

Figure 23

Scrambling sensitivity without the hippocampus: simulation results for Zuo et al. (2020). The violin plots show the magnitude of scrambling sensitivity in the model for the "control group" (the full model) and "patients" (the model without the episodic memory module). Scrambling sensitivity is defined as the reduction of the spatial ISC values in the stimulus-scrambled condition compared to the intact condition (as in Zuo et al., 2020). The result shows that the magnitude of scrambling sensitivity for the hippocampally-lesioned ("patient") model is significantly above zero. This indicates that 1) the activity dynamics of the hippocampally-lesioned model are sensitive to stimulus history and 2) the model's ability to integrate information over time in the continuous setting is not heavily dependent on episodic recall.

The specific goal of this simulation was to model the findings of a recent study by Zuo et al. (2020), which found that a patient with a hippocampal lesion had significant (above zero) scrambling-sensitivity in the DMN. Specifically, Zuo et al. (2020) computed spatial ISC between controls and a hippocampal patient while they were listening to an audiobook. Spatial ISC at a given moment is the linear correlation between the multivoxel patterns of controls and patients. They showed that control-patient spatial ISC was reduced for many DMN regions when the stimulus was scrambled at the level of paragraphs (Zuo et al., 2020) (Figure 22). Additionally, they also found that the level of control-patient ISC for the intact stimulus was quite high (Zuo

et al., 2020), which is consistent with another recent study (Oedekoven et al., 2019). Collectively, these results show that the DMN is capable of integrating information over a fairly long timescale even when the hippocampus has been lesioned. In other words, continuous processing of naturalistic stimuli appears to be possible without heavily engaging episodic memory.

To simulate the results of this study, we used the trained models (n = 30) from *Simulation 2*. We removed the episodic memory module for half of the models to simulate hippocampally-lesioned patients. After functionally aligning these networks (Lu et al., 2018), we presented both a fully intact sequence and its scrambled version to each model. In our simulation, the scrambled event sequences were created by cutting the event sequences in the middle and reversing the presentation order of the two sub-sequences. Then we performed the same analysis as described in Zuo et al. (2020). We found that the level of scrambling sensitivity among the "patient" models is significantly above zero (Figure 23). This result confirms that, without the episodic module, the cortical part of our model is still capable of integrating information over a fairly long timescale, by actively maintaining this information in working memory.

Simulation 8: Integrating information across multiple event segments

As noted earlier, most "real-world" event sequences do not provide all information about the ongoing situation. Moreover, it has been pointed out that humans can use episodic memory to integrate information over many event segments (Griffiths & Fuentemilla, 2020). A recent study by Chang et al. (2020) examined the neural basis of this process with a naturalistic stimulus. In the Chang et al. (2020) study, participants listened to an audiobook narrative that repeatedly alternated between two apparently-distinct storylines, storyline A and storyline B. Eventually, the two storylines converged. In this simulation, we focus on how participants were able to understand ongoing events as the narrative alternated back-and-forth between the A and B storylines.

In the experiment by Chang et al. (2020), the alternating story segments were approximately one minute long, creating a series of one-minute-long "gaps" that participants had to bridge in order to integrate segments relating to the same storyline. Chang et al. (2020) hypothesized that participants bridge these gaps by storing information pertaining to a particular storyline (e.g., storyline A) in episodic memory; later on, when participants return to storyline A,

they can retrieve these storyline-specific episodic memory traces to guide event prediction. As an initial analysis, Chang et al. (2020) examined the similarity between the current multi-voxel pattern and the average patterns associated with storyline A and storyline B in posterior cingulate cortex (PCC; Figure 24A). They found that, when the storyline switched (such as from B to A), the PCC multi-voxel pattern became more similar to the average pattern associated with storyline A. Moreover, the neural difference in the storyline A and B patterns was larger for later segments compared to earlier segments (Chang et al., 2020). Finally, they found that the storyline effect (i.e., the switch in neural patterns when the storyline switched) was positively related to hippocampal-mPFC inter-subject functional correlation (ISFC), a neural signature of episodic recall – these results support the view that reinstating situation models depends on hippocampal-cortical communication, and that the storyline effect is related to episodic recall.

To simulate these results, we tested our model with events from two alternating situations. Concretely, the event sequences alternated between situation A and situation B three times. We will denote the event sequences by A0, B0, A1, B1, A2, and B2. At event boundaries (switch points between A and B), the model encoded an episodic memory. To test if the model could integrate information across blocks, the model was only allowed to observe 60% of the situation feature values during (each of) the first two blocks of each event sequence (e.g., A0, A1): That is, on 60% of time points within each block (randomly selected), a feature value (not previously observed in this block) was observed; on the other 40% of time points, the model was not given an observation (in the third and final block of each event sequence, 100% of the situation feature values were observed, capturing the idea that participants will have observed all of the relevant aspects of the situation by the time they reach the end of the narrative). This arrangement ensured that there would be overlap in observed features between blocks corresponding to a particular situation, but also that the first two blocks did not (on their own) provide a complete situation model. Otherwise, the simulation parameters were largely the same as in *Simulation 2*, as we simply took the models trained in *Simulation 2*, and trained them further on this task with alternating situations.

Mirroring the analysis conducted by Chang et al. (2020), we computed – for each time point around event boundaries (i.e., switches between the two situations) – the pattern similarity

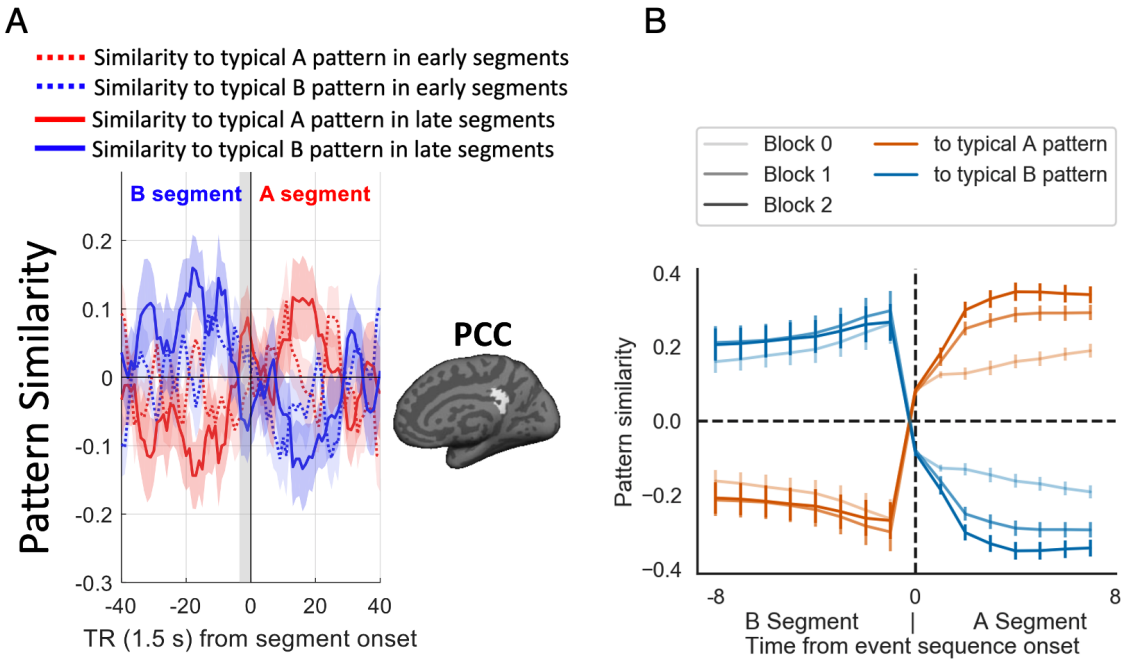

Figure 24

The storyline effect from Chang et al. (2020): data and simulation. Part A reprinted with permission. A) At moments when the storyline switched, the multi-voxel pattern in posterior cingulate cortex (PCC) rapidly became more similar to the average pattern associated with the post-boundary storyline, and this effect was larger for later segments compared to earlier segments (Chang et al., 2020). B) Simulation results: Across an event boundary, the cell state pattern rapidly becomes more similar to the average pattern associated with the post-boundary situation. The pattern-switching effect is stronger for later blocks – this occurs because the situation models encapsulated in episodic memory (for both situation A and situation B) are more complete for later blocks (see Figure 25). Results are averaged across 15 models; errorbar indicates SEM.

of ongoing cell states to the average patterns evoked by situation A and situation B (Figure 24B). We found that, shortly after event boundaries, the pattern of neural activity in the network rapidly becomes more similar to the average pattern associated with the post-boundary situation, and this switching effect is larger for later blocks (block 2 > block 1 > block 0), consistent with the findings from Chang et al. (2020). We also simulated the finding that the amount of episodic recall was correlated with the storyline effect (Chang et al., 2020), by using memory activation as a proxy for recall (similar to the analysis used in *Simulation 6*). We found that memory activation (at time t) is positively correlated with the shift (from time t to time $t+1$) towards the average pattern associated with the ongoing (post-boundary) situation; correlation = 0.275, 95% CI = [0.198, 0.352].

Collectively, these findings suggest that the pattern switch at situation boundaries is caused by episodic recall, and that reinstated memories can rapidly move the network's internal state towards the typical pattern associated with the post-boundary situation. In the model, the storyline effect is larger for later blocks because – each time the model returns to a storyline – it can observe new features and also retrieve (from episodic memory) previously-observed features; then, at the end of the segment, it can store a new episodic memory containing all of these features. Because of this dynamic, episodic memories corresponding to a particular storyline come to encapsulate increasingly complete situation models across blocks, making them more efficient in moving the internal states of the model towards the typical neural representation of that storyline.

To obtain further support for the point that later episodic memories encapsulate more complete situation models, we used MVPA methods to decode the information being actively maintained in the model's working memory at each time point. Specifically, for each feature, we used a logistic regression classifier with L2 penalty to decode if this feature was present in the LSTM cell state at each time point. For the initial block (e.g., A0), we used a ten-fold cross validation procedure to both train and test the classifier on this block. Then we trained the classifier on all of the initial-block data (e.g., A0) and tested the classifier on later blocks (e.g., A1 and A2). The regularization parameter was determined by an inner-loop cross-validation. Figure 25A shows which features were decodable at each time point for the three A blocks (A0, A1 and A2) from an example trial. For the first block, features are only decodable after they are observed

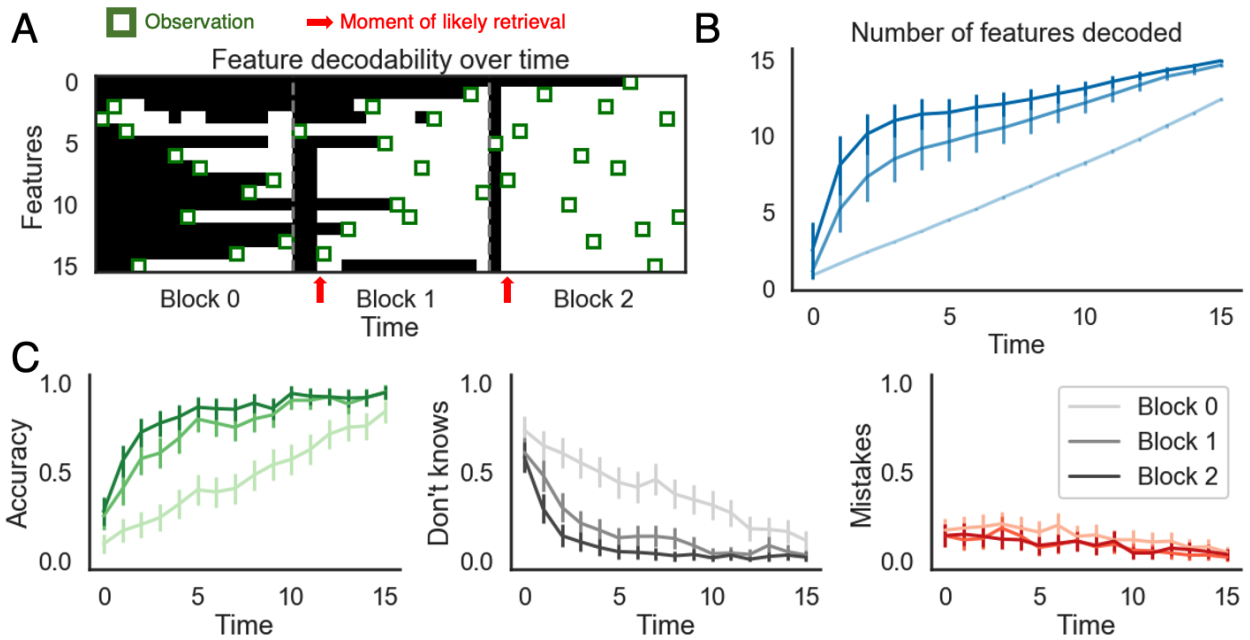

Figure 25

Working memory in later blocks contains a more complete situation model. Panel A is a single trial demo. The heatmap shows which features were decodable (in the cell state) at each time point for all three blocks from situation A (white = correctly decoded, black = not correctly decoded). For later blocks, more features can be retrieved from episodic memory, resulting in more features being decodable before they were observed (moments of likely retrieval from episodic memory are highlighted with red arrows on the plot). Panel B shows that, for later blocks, there are more decodable features on average. Panel C shows the prediction accuracy of the model across blocks. During the first block, the model has no relevant stored episodic memories, so behavioral performance increases linearly as the event sequence unfolds. Prediction performance in the second block is better, because the model has a stored episodic memory pertaining to the ongoing situation. The additional improvement from the second to the third block suggests the model can stitch situation features together across event segments with partial information to produce episodic memories with a more complete situation model. For panels B and C, results are averaged across 15 models; errorbar indicates SEM.

(observations are marked with green boxes); for later blocks, more features are decodable before observation, indicating that the model is filling in these features using episodic recall (Figure 25A, B). This "filling in" of features boosts the model's performance: For later event segments, the model shows higher levels of accurate prediction and lower levels of mistakes and "don't know" responses (Figure 25C). Overall, these results demonstrate our model is capable of integrating information over many event segments with partial information, allowing it to form a more complete situation model.

Simulation 9: Schema-consistent behavior when there is a prototypical event

It is well known that people's memories of events are shaped by schematic knowledge of how these events usually unfold, resulting in improved memory for schema-consistent features of those events and also (in cases where the schema is violated) false recall of schema-consistent features (Alba & Hasher, 1983; Bartlett, 1932). For example, in a classic study, participants were asked to wait in the experimenter's office, and they were subsequently asked to recall details of that room (Brewer & Treyens, 1981). Participants' recall was more accurate for aspects of the room that were more consistent with their expectations about academic offices (Brewer & Treyens, 1981), and participants falsely reported items that are typically in an academic office but were not there in the experiment (Brewer & Treyens, 1981).

In the simulations preceding this one, we imposed a form of schematic structure by teaching the model about which events could occur at which time points (i.e., the "columns" of Figure 26). This is akin to teaching the model that, for a three-course meal, we would expect soup, a main course, and finally a dessert (in that order). However, *within* a particular time point, the marginal probabilities of the events that were "allowed" at that time point were equated – put another way, none of the events were more prototypical than any of the other events. In this simulation, we allowed for some events to be more prototypical (i.e., occur more often) than other events that could occur at that time point (e.g. allowing P(cheesecake) to be much greater than other alternatives for dessert). This modification allows us to assess how this kind of schematic knowledge interacts with episodic memory; in particular, it allows us to assess whether our model shows the kinds of schema-consistent memory biases noted above. Furthermore, the model allows

us to evaluate different ideas about *why* these memory biases might occur – e.g., in cases where the model falsely remembers a prototypical event (when in fact the true event was non-prototypical), is it because the model actually activated the prototype in working memory, or is it because of a bias to guess the prototypical event in cases where the model is unsure about what really happened?

Additionally, the model allows us to explore the effects of schema strength on recall policy (i.e., the setting of the input gate). Intuitively, when there is a strong schema, there is less of a need to rely on episodic memory – in the limiting case, if the schematic event occurs in every sequence, the model will learn to predict this event every time and there is no need to consult episodic memory. As discussed below, the model shows this effect, but the effects of schema strength on recall policy are more complex when the regularity is not perfect (i.e., when the schematic event occurs more than other events at that time point, but the schema is sometimes violated).

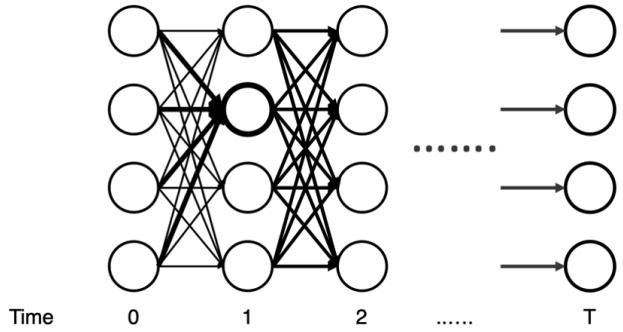

Figure 26

Event graph with prototypical events. Each node is an event. Each edge is a transition. The edge thickness denotes the transition probability. The probability over emitting edges sums to one. The figure shows a graph with prototypical events: When taking a sample path from this graph, the bolded node is much more likely to be visited than the two other nodes at the same time point. In our simulation, half of the time points have a prototypical event (schematic time points) and the other time points do not have a protoytpical event (non-schematic time points).

To explore these issues, we ran simulations where, for half of the time points (*schematic* time points), one of the events at that time point happened with higher probability than the

other events. For the other half of the time points (*non-schematic* time points), all of the events associated with that time point were equally probable (given that there were four possible events at each time point, the probability of each event was .25). Schematic and non-schematic time points were arranged in an alternating fashion (for half of the models, even time points were schematic and odd time points were non-schematic, and the opposite was true for the other half of the models). For schematic time points, we manipulated the the strength of schematic regularity in the environment by manipulating the probability of the "prototypical" event. We tried schema strength values from 0.25 (baseline) to 0.95 in steps of 0.10. We implemented this regularity by manipulating the edge probabilities in the event graph. Figure 26 illustrates a hypothetical event graph; in the figure, time point 1 (a schematic time point) has a typical event, whereas all events are equally likely for time point 0 and time point 2 (non-schematic time points). When taking a sample path from this graph, the bolded node at time point 1 is much more likely to be traversed than the three alternative nodes at the same time point. In an event sequence, if the event (node) at schematic time point t matches the prototype, we say this is a *schema-consistent* event; if the event is atypical (i.e., a prototypical event exists for that time point, but this event is not it), we say that this is a *schema-violated* event. Finally, in this simulation, we used a fixed penalty value (0, 2, and 4) for both training and testing (i.e., some models were trained and tested with penalty 0, some models where trained and tested with penalty 2, etc). We trained 15 models for each combination of schema strength and penalty. After training, we examined how the model behaved during schematic and non-schematic time points, as a function of schema strength (for the schematic time points).

We found that our model shows a tendency to make schema-consistent responses (Figure 27A): When the current time point has a prototypical event and the prototypical event occurred (i.e., it was schema consistent), the model shows an enhanced tendency to be correct. On the other hand, if the prototypical event did not occur (i.e., the schema was violated), the model tends to make schema-consistent mistakes. This pattern resembles the classic false memory effect (Alba & Hasher, 1983; Bartlett, 1932). In the model, both of these effects (enhanced correct responses for schema-consistent events, and schema-consistent mistakes for schema-violated events) increase in a graded fashion as the level of schema strength increases.

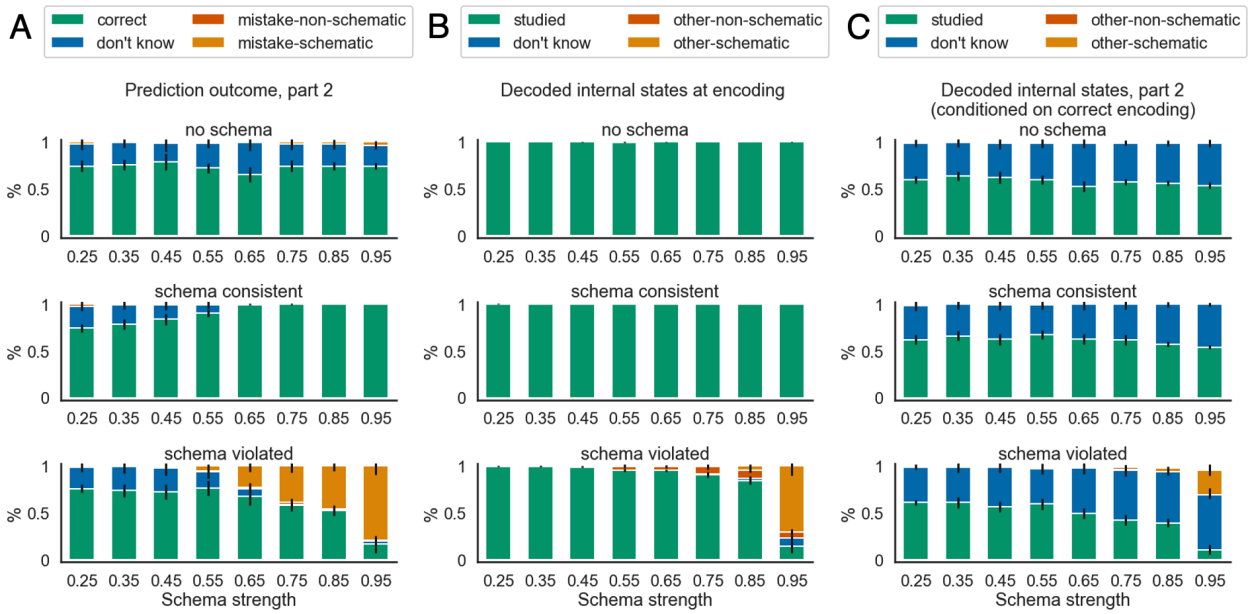

Figure 27

The model shows a schema-consistent false memory effect. These plots show the effect of increasing schema strength on prediction behavior, decoded internal states at encoding, and decoded internal states during part two. A) For time points with no schema, increasing schema strength has no effect on behavior. For time points with a schema, increasing schema strength boosts correct responses (shown in green) if the event is schema consistent (i.e., the prototypical event happened) and boosts schema-consistent mistakes (shown in orange) if the schema was violated (i.e., the prototypical event did not happen). B) To compute encoding accuracy, we compared the decoded feature values on the final time step of part one (when episodic encoding took place) to the ground-truth (observed) feature values. We found that encoding is highly accurate, unless schema strength is extremely high (at .95) and the schema was violated; in this circumstance, the model frequently intrudes the prototype ("other-schematic" errors, shown in orange). C) Decoding results during part two (for trials where there was no encoding error during part one) showed a similar pattern: In the schema-violated condition, the model only intrudes the prototype into working memory when schema strength is extremely high. Results are averaged across 15 models; errorbar indicates SEM.

To better understand this phenomenon, we used multivariate pattern analysis (MVPA) to decode the internal state of the network (these MVPA analyses were conducted on our simulations where penalty = 2). As noted above, one possible explanation of the schema-consistent memory effect is that the model is activating the prototype in working memory even when it has not been observed. To analyze encoding accuracy, we decoded the model's internal states on the final time step of part one, since that is when episodic encoding took place. Then we compared decoded internal states with the ground-truth (observed) situation feature values to determine if a decoded internal state corresponded to the studied feature value (studied), the "don't know" state, or some other feature value that was prototypical (other-schematic) or not prototypical (other-non-schematic). We found that, when there is no schema or when the situation features are schema consistent, the decoded internal states almost always match the studied situation feature value (Figure 27B). However, when the schema was violated (i.e., the prototypical event did not happen) and schema strength is extremely high (schema strength = .95), the model makes a large number of encoding errors, substituting the prototype for the actually-studied value (Figure 27B). Overall, these results suggest that some schema-consistent memory errors may be caused by encoding failure (when schema strength is high and the schema was violated), but – apart from this particular circumstance – encoding is very accurate.

Next, we decoded internal states during part two of the sequence (excluding features that were mis-encoded during part one). Similarly to what we found during part one, we found that the model sometimes actively intrudes the prototype in working memory when the schema was violated and schema strength is maximal, but otherwise these kinds of errors are extremely rare (Figure 27C).

These results, taken together with the part one decoding results, pose a puzzle: The model appears to be actively intruding the prototype feature into working memory only at the highest levels of schema strength, but – behaviorally – the model shows a more graded schema-consistent memory effect that is evident at much lower levels of schema strength (Figure 27A). This suggests that some other factor must be contributing to the schema-consistent memory effect. As shown in Figure 28, this other factor is *decision bias* – the figure shows the proportions of different response outcomes for schematic time points (schema consistent or schema violated), conditioned on the

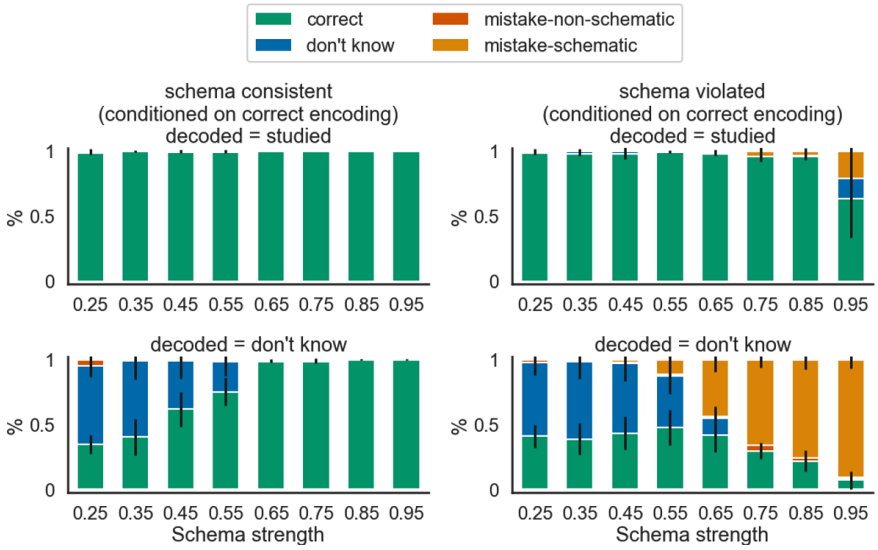

Figure 28

The model makes schema-consistent predictions when it is uncertain. The second row of the figure shows that, if the model's internal state is "don't know", increasing schema strength boosts the odds of the model giving a prototypical response; this leads to 1) an increase in correct responding when the feature's value is schema-consistent (prototypical) and 2) an increase in schema-consistent errors (mistakenly predicting that the prototypical event will happen) when the feature's value is schema-inconsistent (i.e., when the prototype was violated). Results are averaged across 15 models; errorbar indicates SEM.

decoded states. Here we are focusing on the response outcome when the decoded states are either the studied feature value or "don't know". These results show that increasing schema strength boosts the model's tendency to guess the prototypical feature when its internal state is "don't know". This tendency has the effect of 1) increasing correct responses when the event is schema consistent and 2) increasing schema-consistent mistakes when the schema was violated. Importantly, we want to note that this particular pattern of schema-consistent memory effects is dependent on the parameters that we used; future work could explore what happens in other scenarios – for example, in our current (simplified) modeling paradigm, the model does not have any stored episodic memories other than the memory for part one of the sequence. If, instead, we allowed the model to store a large number of "pre-experimental" episodic memories of schema-consistent events, the model might show a much stronger tendency to intrude prototypical

events (and actively represent them in working memory) on schema-violated trials.

Our final goal for this simulation was to use the model to explore how schematic knowledge interacts with recall policy: Does the existence of a prototypical event at a particular time point make the model more or less likely to consult episodic memory when queried about that time point? The results of this analysis for penalty = 2 (the penalty value used in the MVPA analyses above) are shown in Figure 29 (middle panel), which plots input gate as a function of schema strength. The first thing to note about these results is that, for high levels of schema strength, recall is suppressed for schematic time points (i.e., time points with a prototypical event) and elevated for non-schematic time points (i.e., time points where there was not a prototypical event). The former finding (suppression of recall at time points where there is a strong prototype) can be explained in terms of the idea, mentioned earlier, that high-schema-strength events are almost fully predictable without episodic memory, and thus there is no need to recall. The latter finding (enhanced recall at non-schematic time points, when schema strength is high overall) can be explained in terms of the idea that schema-consistent features tend to be shared by both target and lure memories and thus are not diagnostic of which memory is the target; in this situation, the only way to distinguish between targets and lures is to recall non-schematic features, which is why the model tries extra-hard to recall them.

Interestingly, the model shows the opposite pattern of effects when schema strength = .55 or .65: Recall is enhanced for schematic time points and suppressed for non-schematic time points. This reversal can be explained as follows: When schema strength = .55 or .65, the model has started to build up a tendency to guess the schema-consistent (prototypical) answer (see Figure 28), but it is also going to be wrong about 1/3 of the time when it guesses the schema-consistent answer, incurring a substantial penalty. To counteract this tendency to make wrong guesses, the model needs to try extra-hard to recall the actual feature value for schematic time points (which is why the input gate increases for these time points) – and if the model is doing more recall in response to schematic features, it needs to do somewhat less recall in response to non-schematic features (which is why the input gate goes down for these features). As schema strength increases beyond .65, the model will be wrong less often when it guesses the schema-consistent answer, so there is less of a need to counteract wrong guesses with episodic

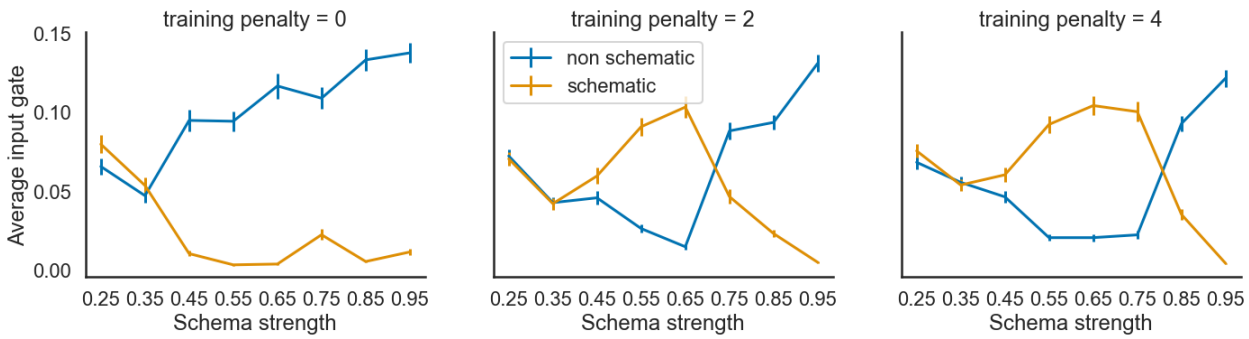

Figure 29

Input gate as a function of schema strength and penalty, plotted separately for schematic and non-schematic time points. The figure shows input gate (averaged over time) as a function of schema strength for penalty = 2 (the value used in the schema simulations described above; middle panel) and also lower and higher penalty values (penalty = 0 on the left; penalty = 4 on the right). For penalty = 0, input gate for schematic time points decreases monotonically as a function of schema strength, and input gate for non-schematic time points shows the complementary (increasing) pattern. For higher levels of penalty, input gate for schematic time points shows an inverted-U pattern as a function of schema strength, increasing then decreasing, and input gate for non-schematic time points shows the complementary (U-shaped) pattern. See text for explanation of these patterns. Results are averaged across 15 models; errorbar indicates SEM.

recall – this makes it safe for the model to reduce the input gate for schematic time points at higher levels of schema strength (as described above).

This account (i.e., that the model is consulting episodic memory at schema strength = .55 or .65 to avoid penalties caused by incorrect habitual responding) leads to the prediction that reducing the penalty should eliminate this effect – if there is no cost to incorrect habitual responding, the model does not have to work as hard to overcome this habitual responding with recall. To test this, we ran a variant of the simulation with penalty = 0 (Figure 29, left-hand panel). The results of this simulation confirm our prediction: When penalty = 0, the model's tendency to consult episodic memory on schematic time points declines monotonically with schema strength. We also ran a variant of the simulation with a higher penalty value (penalty =

4). We expected that this would enhance the model's tendency to consult episodic memory on schematic time points for moderate levels of schema strength, because the costs of incorrect habitual responding would be even higher; the results of this simulation (shown in Figure 29, right-hand panel) confirm this prediction. Overall, these simulations highlight the complex ways that schematic regularities can shape recall policy.

Empirical prediction 4 – the influence of schema strength and penalty on recall policy

To test the model's predictions (shown in Figure 29) about how schema strength affects reliance on episodic memory, the paradigm described in *Empirical Predictions 1 and 2* can be modified so that some time points in the sequence have prototypical events and some do not, and we can manipulate the strength of these regularities (i.e., the probabilities of the prototypical events) and the penalty for prediction errors, mirroring the structure of *Simulation 9.* Use of recall can be measured with hippocampal-cortical ISFC, following J. Chen et al. (2016). For very high levels of schema strength, we expect the neural signature of recall to be larger for nonschematic than schematic time points, regardless of penalty. For more moderate levels of schema strength, we expect that the level of recall for schematic vs. non-schematic time points will depend on the penalty. When the penalty is low, we expect more recall for non-schematic than schematic time points. When the penalty is high, we expect that opposite pattern – more recall for schematic than non-schematic time points.

General discussion

Most of what we know about episodic memory has come from experiments where participants were told when encoding and retrieval should take place and what information should be encoded and retrieved (e.g., learning and recalling lists of random word pairs), and most computational models of human memory have focused on explaining findings from these experiments (for a review, see Norman et al., 2008). In this work, we instead explored how episodic memory contributes to event understanding; a key feature of this scenario is that participants are given temporally-extended sequences as inputs, and they have to determine on their own what information to hold in working memory, when to store snapshots of the contents

of working memory, and when to retrieve these snapshots. We formalized event understanding as a situation-dependent event prediction task, where event sequences are samples from an underlying event schema graph conditioned on features of the current situation (i.e., a situation model). We trained neural networks with episodic memory to obtain approximately optimal solutions to this task; a key feature of the model is that it *learned its own policy* for when to consult episodic memory (via an adjustable gate) in order to maximize reward. Our analyses of the network policies for event prediction, episodic retrieval, and episodic encoding, optimized under different environmental regimes, led to normative accounts for a wide range of empirical findings from experiments using naturalistic stimuli.

The most fundamental result from our simulations was that the model learned to use episodic memory to support event prediction. Episodic memory is useful in our model because situations (configurations of features that determine transitions between events) sometimes reoccur. If the system succeeds in retrieving a stored situation model that matches the current situation, this will improve its ability to predict upcoming events. The risk of retrieving from episodic memory is that, if the system retrieves a stored situation model that mismatches the current situation, this will cause it to make incorrect predictions; this risk is especially high at the beginning of an event sequence, since recent observations do not provide enough information to disambiguate the target memory versus lures.

Our simulations found that the model learns policies that appropriately balance the benefits and risks of episodic recall, modulating recall based on the penalty for incorrect predictions (*Simulations 1, 2, 3, and 9*), the confusability of memory traces (*Simulation 3*), and its own uncertainty (*Simulation 1*: deciding not to risk recalling when it was confident that it knew what would happen next). We also showed in *Simulation 4* that the model is able to leverage a familiarity signal (indicating whether or not relevant episodic memories were available) to modulate its recall policy. These results allowed us to explain some existing findings – notably, that showing a familiar stimulus increases the extent to which information is subsequently retrieved from the hippocampus (Duncan et al., 2012; Duncan et al., 2019; Duncan & Shohamy, 2016; Hasselmo & Wyble, 1997; Patil & Duncan, 2018), and that inserting a 24-hour gap between the halves of a TV episode increases the extent to which hippocampus and cortex communicate at

the start of the second half (because of increased uncertainty induced by the 24-hour gap) (J. Chen et al., 2016; see also *Simulation 6*). Several of the aforementioned predictions (e.g., regarding penalty effects) remain to be tested, and we suggested specific ways to these these predictions (see *Empirical Predictions 1 and 2*).

In *Simulation 5*, we provided a normative account for the empirical observation that episodic encoding seems to preferentially occur at event boundaries (Baldassano et al., 2017; Ben-Yakov & Dudai, 2011; Ben-Yakov & Henson, 2018). In our simulation, we found that models that selectively encode at event boundaries perform the best. This is because episodic memories formed at event boundaries contain a more fully-specified situation model, which makes subsequent disambiguation during recall easier, reducing the probability of recalling lures. We also described a novel way to test this hypothesis going forward (see *Empirical Prediction 3*).

Most fMRI studies of memory in naturalistic situations use pattern similarity as a dependent measure (either inter-subject correlation – similarity in neural patterns observed across different individuals perceiving or remembering a stimulus – or similarity across conditions within individuals). In *Simulations 6, 7, and 8*, we showed that the model can account for key findings in this space: In *Simulation 6*, we showed that the model can explain the finding from J. Chen et al. (2016) that episodic memory retrieval fills in gaps in participants' understanding of a movie, boosting inter-subject correlation. In *Simulation 7*, we showed that – when the ongoing sequence of events is not interrupted by an event boundary – the cortical model is able to maintain a representation of recent events in working memory, mirroring recent findings obtained by Zuo et al. (2020) in an amnesic patient. In *Simulation 8*, we showed how the model can use episodic memory to bridge interruptions in a narrative to assemble a coherent situation model, thereby explaining recent findings from Chang et al. (2020). The dynamics shown in *Simulation 8* fit with prior work showing hippocampal contributions to memory integration at encoding (Backus et al., 2016; M. L. Schlichting & Preston, 2015, 2016; M. L. Schlichting et al., 2014; Shohamy & Wagner, 2008; Zeithamova, Dominick, et al., 2012; Zeithamova & Preston, 2010; Zeithamova, Schlichting, et al., 2012), and also with the idea that hippocampus plays a particularly important role in combining information across event boundaries (Clewett et al., 2019; Cohn-Sheehy et al., 2020; DuBrow & Davachi, 2014, 2016; Ezzyat & Davachi, 2014; Griffiths & Fuentemilla, 2020).

Another contribution of these simulations (in particular, *Simulations 6 and 7*) is that they provide an proof-of-concept example for how to model inter-subject neuroimaging data with neural networks. Inter-subject analyses usually require aligning different subjects into a common space, with anatomical or functional alignment methods (Nastase et al., 2019). In this work, we used the shared response model (P.-H. Chen et al., 2015; Lu et al., 2018), a method commonly used for functionally aligning neuroimaging data across subjects, to align hidden states across neural networks. Future studies can use this approach to model other empirical results that have used inter-subject analyses.

The final simulation (*Simulation 9*) explored what happens when we introduce prototypical states (i.e., states that occur more often than other states that are possible at that time point). We were able to simulate classic schema-dependent memory biases (i.e., increased correct recall of prototypical features; false recall of prototypical features in cases where the studied feature was not prototypical). The model also predicts that schema strength should affect the model's reliance on episodic memory in complex ways that interact with the penalty on incorrect predictions, boosting the use of recall in some situations and reducing it in others (see *Empirical Prediction 4* for discussion of how to test these predictions). More broadly, this simulation shows how the model can be used to explore interactions between three distinct memory systems: semantic memory (instantiated in the weights in cortex), working memory (instantiated in the gating policy learned by the cortical LSTM module, allowing for activation at one time point in cortex to influence activation at subsequent time points), and episodic memory. In the past, modelers have focused on these memory systems in isolation (see, e.g., Norman et al., 2008), in part because of a desire to understand the detailed workings of the systems, but also because of technical limitations: Until very recently, the technology did not exist to automatically optimize the performance of networks containing episodic memory, so researchers interested in simulating interactions between episodic memory and these other systems were put in the position of having to do time-consuming (and frustrating) hand-optimization of the models. Here, we leverage recent progress in the artificial intelligence literature on memory-augmented neural networks (Graves et al., 2016; Pritzel et al., 2017; Ritter et al., 2018; Wayne et al., 2018) that makes it possible to automatically optimize the use of episodic memory and its interactions with

other memory systems. This technical advance has opened up a new frontier in the cognitive modeling of memory (Collins, 2019), making it possible to address richer "naturalistic memory" scenarios that involve interactions between prior knowledge, active maintenance, and episodic memory.

Relation to other models

Memory-augmented neural networks with a differentiable neural dictionary

Conceptually, the episodic memory system used in our model is similar to recently-described memory-augmented neural networks with a differentiable neural dictionary (DND) (Pritzel et al., 2017; Ritter, 2019; Ritter et al., 2018). In these models, the data structure of the episodic memory system is dictionary-like: Each memory is a key-value pair. The keys define the similarity metric across all memories, and the values represent the content of these memories. For example, one can use the LSTM cell state patterns as the keys and use the final output of the network as the values (Pritzel et al., 2017); note that, in our model, the cell state of the cortical network serves as both the key and the value. The work by Ritter et al. (2018) is particularly relevant as it was the first paper (to our knowledge) to use the DND for cognitive modeling and – as such – served as a major inspiration for the work presented here (see also Botvinick et al., 2019). The way that our model uses the DND mechanism is quite similar to how it was used in Ritter et al. (2018); in particular, we took from the Ritter et al. (2018) paper the idea that the cortical network learns to control a "gate" on episodic recall via reinforcement learning.

However, there are also some meaningful differences between our model and the model used by Ritter et al. (2018). One difference is that our model's computation of which memories are retrieved (given a particular retrieval cue, assuming that the "gate" on retrieval is open) is more complex. Ritter et al. (2018) used a one-nearest-neighbor matching algorithm during recall, whereby the stored memory with the highest match to the cue is selected for retrieval (assuming that the gate is open). By contrast, memory activation in our model is computed by a competitive evidence accumulation process, in line with prior cognitive models of retrieval (e.g., Polyn et al., 2009; Sederberg et al., 2008). While we did not explore the effects of varying the

level of competition in our simulations, having this as an adjustable parameter opens the door to future work where the model learns a policy for setting competition in order to optimize performance (just as it presently learns a policy for setting the input gate).

Another mechanistic difference between our model and the Ritter et al. (2018) model is our addition of the "don't know" output unit, which (when selected) allows the model to avoid both reward and punishment. The addition of this unit incentivizes the model to explicitly represent its own uncertainty about feature values (as shown in Figure 3, the model uses a particular region of hidden-state space, characterized by a low activity norm, to represent when a feature's value is unknown). Relatedly, in *Simulation 1.1*, we also showed that episodic recall in our model is driven by model's uncertainty about the upcoming event – the model recalls more (by opening the retrieval gate) when the queried information is not already being maintained in working memory.

Apart from the technical differences noted above, the main difference between our modeling work and the work done by Ritter et al. (2018) relates to the application domain (i.e., which cognitive phenomena were simulated). Our modeling work in this paper focused on how episodic memory can support incidental prediction of upcoming events (i.e., when there is no explicit demand for a decision). By contrast, Ritter et al. (2018) focused on how episodic memory can be used to support performance in classic decision-making tasks, such as bandit tasks and maze learning, that have been extensively explored in the reinforcement learning literature. In the bandit task used by Ritter et al. (2018), the model sees a sequence of bandits and makes decisions about which lever to pull; when the model encounters a previously-seen bandit, it can retrieve an episodic memory about this bandit, allowing it to avoid re-exploration and start exploiting immediately (Ritter et al., 2018). In the contextual water maze task (Morris, 1984) used by Ritter et al. (2018), the model needs to solve a sequence of maze problems as quickly as possible; when the model is placed in a previously-seen maze, the model can use episodic memory to retrieve the goal location and pick the shortest path (Ritter et al., 2018). Finally, Ritter et al. (2018) tested the model with an episodic two-step task (Vikbladh et al., 2019). The classic two-step task assesses where the agent is along the spectrum from model-based to model-free decision making (Daw et al., 2011). In their simulations of the episodic two-step task, Ritter et al. (2018) found that the model used episodic memory to solve the task, and the relative contribution

of different strategies was qualitatively similar to human empirical results. For all of these simulations, Ritter et al. (2018) also showed that their model outperformed other models that did not incorporate the episodic memory module.

The structured event memory (SEM) model

Another highly relevant model is the structured event memory (SEM) model developed by Franklin et al. (2020). Like our model, SEM uses RNNs to represent its knowledge of schemas (i.e., how events typically unfold). Also, like our model, SEM records episodic memory traces as it processes events. However, there are several key differences between our model and SEM. First, whereas our model uses a single RNN to represent a single (contextually parameterized) schema, SEM uses multiple RNNs that each represent a distinct schema for how events can unfold. Building on prior work on nonparametric Bayesian inference (Aldous, 1985; Anderson, 1991; Pitman, 2006) and latent cause modeling (Gershman et al., 2010; Gershman et al., 2015), SEM contains specialized computational machinery that allows it to determine which of its stored schemas (each with its own RNN) is relevant at a particular moment, and also when it is appropriate to instantiate a new schema (with its own, new RNN) to learn about ongoing events. This inference machinery allows SEM to infer when event boundaries (i.e., switches in the relevant schema) have occurred, and the Franklin et al. (2020) paper leverages this to account for data on how people segment events. Our model lacks this inference machinery, so we need to impose event boundaries by fiat, as opposed to having the model identify them on their own.

Another major difference between the models relates to how episodic memory is used. The focus of our modeling work in this paper is on how episodic memory can support online prediction. By contrast, in SEM, episodic memory is not used at all for online prediction – online prediction is based purely on event schemas stored in the RNNs. The sole use of episodic memory in the Franklin et al. (2020) paper is to support reconstruction of previously-experienced event sequences. Specifically, in SEM, each time point leaves behind a noisy episodic trace; the Franklin et al. (2020) paper shows how Bayesian inference can combine these noisy stored episodic memory traces with stored knowledge about how events typically unfold (in the RNNs) to reconstruct an event sequence. Effectively, SEM uses knowledge in the RNNs to “de-noise” and fill in gaps in the

stored episodic traces. The Franklin et al. (2020) paper uses this process to account for several findings relating to human reconstructive memory.

Future directions and limitations

Extensions to the modeling framework

On the modeling side, our work can be extended in several different ways. As noted above, our model and SEM have complementary strengths: SEM is capable of storing multiple schemas and doing event segmentation, whereas our model only stores a single schema and we impose event boundaries by hand; our model is capable of using both episodic memory and its schematic knowledge in cortex to do online prediction, whereas SEM only uses its schematic knowledge. It is easy to see how these complementary strengths could be combined into a single model: By adding SEM's ability to do multi-schema inference to our model, we would be able to simulate event segmentation *and* explore the role of episodic memory in event prediction.

Another limitation of the current model is that the encoding policy is not learned. In *Simulation 5*, we trained models with different encoding policies and compared their performance. We observed that models that selectively encode episodic memories at event boundaries perform the best, which provides a normative account of why humans seem to encode episodic memories preferentially at event boundaries (Ben-Yakov & Henson, 2018). Going forward, we would like to develop models that learn when to encode through experience (instead of having to impose encoding policies by hand). Our results show that encoding at event boundaries is the reward-maximizing solution, which suggests that encoding at event boundaries should be a learnable behavior using a reinforcement learning framework. However, in practice, learning an encoding policy is challenging because it involves long-range temporal credit assignment – an extensive amount of time can elapse between deciding to take an episodic memory snapshot and observing the consequence of taking that snapshot (in terms of its effects on prediction error and reward). For example, taking an episodic memory snapshot today when you go to a particular restaurant could potentially affect your ability to make predictions a year later when you return to that restaurant.

Importantly, the key factor shaping encoding policy may not be event boundaries *per se*,

but rather the level of surprise. Zacks and colleagues have found a strong relationship between surprise and event segmentation (Zacks et al., 2011; Zacks et al., 2007; for a recent example see Antony et al., 2021), and numerous studies have found improved subsequent memory for events that are themselves surprising (Alba and Hasher, 1983; Ranganath and Rainer, 2003) or are accompanied by surprise (e.g., Rouhani et al., 2018; Rouhani et al., 2020). Relatedly, a recent paper by Bein et al. (2020) found that prediction error (which is closely related to surprise) biased hippocampal dynamics towards encoding, and other work has found that prediction error boosts memory (e.g., Greve et al., 2017; see also Henson and Gagnepain, 2010). In addition to surprise (and prediction error), recent work by Sherman and Turk-Browne (2020) suggests that *predictive certainty* may play a role in shaping encoding policy: They found that stimuli that trigger strong predictions (i.e., high certainty about upcoming events) are encoded less well. In keeping with this point, Bonasia et al. (2018) found that, during episodic encoding, events that were more typical (and thus were associated with more predictive certainty, and less surprise) were associated with lower levels of medial temporal lobe (MTL) activation. Intuitively, it makes sense to focus episodic encoding on time periods where there is high surprise and low predictive certainty – if events in a sequence are unsurprising and associated with high predictive certainty, this means that existing (cortical) schemas are sufficient to reconstruct that event, and no new learning is necessary. Conversely, if events in a sequence do not follow a schema (leading to uncertainty) or violate that schema (leading to surprise), the only way to predict those events later will be to store them in episodic memory. Future work can explore whether a model that has "metacognitive" access to surprise and certainty can learn to leverage one or both of these factors when deciding when to encode (our present model is a good place to start in this regard, as we have already demonstrated its ability to factor certainty into its retrieval policy).

Another major simplification in our model is that it encodes each episodic memory as a distinct entity in the DND mechanism (i.e., old memories are never overwritten or updated). This view conflicts with the growing literature on memory reconsolidation, which suggests that memory reminders can result in participants accessing an existing memory and then updating that memory, rather than forming a new memory outright (Dudai, 2009; Dudai & Eisenberg, 2004; Hardt et al., 2010; S.-H. Wang & Morris, 2010). This could be implemented in our model

by having the model try to retrieve before it encodes a new memory; if it succeeds in retrieving a stored memory above a certain threshold level of activation, the model could update that memory in the DND rather than creating a new memory. In future work, we plan to implement this mechanism and use it to simulate memory reconsolidation data.

Going forward, we also hope to explore more biologically realistic episodic memory models (e.g., Ketz et al., 2013; Norman and O'Reilly, 2003; Schapiro et al., 2017). Using a more biologically realistic hippocampus could affect the model's predictions (e.g., if memory traces were allowed to interfere with each other during storage) and it would also improve our ability to connect the model to neural data on hippocampal codes and how they change with learning (e.g., Brunec et al., 2020; Chanales et al., 2017; Duncan and Schlichting, 2018; Favila et al., 2016; Hulbert and Norman, 2015; Kim et al., 2017; Ritvo et al., 2019; Schapiro et al., 2012; Schapiro et al., 2016; M. L. Schlichting et al., 2015; Stachenfeld et al., 2017; Whittington et al., 2020). Similarly, using a more biologically detailed cortical model (separated into distinct cortical sub-regions) could help us to connect to data on how different cortical regions interact with hippocampus during event processing (e.g., Barnett et al., 2020; Cooper et al., 2020; Gilboa and Marlatte, 2017; Preston and Eichenbaum, 2013; Ranganath and Ritchey, 2012; Ritchey and Cooper, 2020; van Kesteren et al., 2012). We have opted to start with the simplified episodic memory system described in this paper both for reasons of scientific parsimony and also for practical reasons – adding additional neurobiological details would make the model run too slowly (the current model takes on the order of hours to run on standard computers; adding more complexity would shift this to days or weeks).

Just as our model contains some key simplifications, the task environments used in this paper are also relatively simple and do not capture the full richness of naturalistic event sequences. Some recent studies have explored event graphs with more realistic structure (C. Chen et al., 2019; Elman & McRae, 2019). The fact that our model can presently only handle one schema substantially limits the complexity of the sequences it can process; adding the ability to handle multiple schemas (as discussed above) will help to address this limitation. Also, natural events unfold over multiple timescales. For example, going to the parking lot is an event sequence that involves finding the key, getting to the elevator, etc., but this can be viewed as part of a

higher-level event sequence, such as going to an airport. In our simulation, events only have one timescale. In general, introducing additional hierarchical structure to the stimuli would enrich the task demands and lead to interesting modeling challenges. For now, we have avoided more complex task environments for computational tractability reasons, but – as computational resources continue to grow – we hope to be able to investigate richer and more realistic task environments going forward.

Another way to extend this work is to simulate other kinds of tasks besides event prediction. In recent years, a rich literature has emerged showing that, in addition to recalling past events and predicting upcoming events (on-line), episodic memory contributes to simulating hypothetical scenarios (see, e.g., Addis, 2018; Addis et al., 2009; Addis et al., 2007; Beaty et al., 2018; Hassabis et al., 2007), including counterfactual scenarios (De Brigard et al., 2013). For example, even if you have never been to a sushi restaurant on a boat, you can use your prior episodic memories from sushi restaurants and boats to construct a detailed mental simulation of how this experience would unfold. Much of the same machinery that we use here to simulate event prediction could be re-used to simulate this kind of episodic future simulation task – instead of predicting one step ahead, the model could be used to predict an entire hypothetical sequence. However, we expect that some changes would also have to be made to the model, in particular to the objective function used during learning. When optimizing the model for event prediction (as described in this paper) the objective function is based on rewards for correct predictions and penalties for incorrect predictions. In the episodic future simulation scenario, prediction error can not be used to evaluate the "goodness" of the simulation, because the participant does not receive feedback from the world (i.e., I am not told whether my sushi-restaurant-on-a-boat simulation was correct or not). Rather, one might expect that the proper objective function for simulation would focus on completeness (i.e., the more details that are filled in, the better the simulation) and consistency (i.e., the more "stable" the contents of the simulation are, the better; you don't want to start simulating a sushi meal on a boat and end up simulating a meal on land).

Lastly, although we have focused on cognitive modeling in this paper, we think that some of our results have implications for machine learning more broadly. For example, most memory-augmented neural networks used in machine learning encode on each time step (Graves

et al., 2014; Graves et al., 2016; Pritzel et al., 2017; Ritter et al., 2018). Our results from *Simulation 5* provide initial evidence that taking episodic "snapshots" too frequently can actually harm performance – storing snapshots partway through an event can lead to retrieval of irrelevant memories and (through this) misprediction. Future work can explore the circumstances under which more selective encoding policies might lead to improved performance on machine learning benchmarks.

Summary

The modeling work presented here builds on a wide range of research showing that episodic memory is a resource that the brain can flexibly draw upon to solve tasks (see, e.g., Bakkour et al., 2019; Biderman et al., 2020; Palombo et al., 2019; Palombo et al., 2015; Shohamy and Turk-Browne, 2013). This view implies that, in addition to studying how episodic memory works in situations where memory is explicitly being tested, it is also valuable to study how episodic memory is used in the wild, outside of memory experiments, when we are simply trying to understand the world, predict what happens next, and make good decisions. To engage with this new, naturalistic memory literature, we have leveraged advances in AI that make it possible for models to learn a policy for how to use episodic memory that maximizes reward – our simulations provide a way of predicting how episodic memory should be deployed (i.e., the "optimal policy") to anticipate what comes next, as a function of the properties of the learning environment (e.g., how confusable are the situations being experienced; how bad is it to make an incorrect prediction). While our understanding of episodic memory in the wild is still meager compared to our understanding of episodic memory in the lab, our hope is that this model (and others like it, such as the model by Ritter et al., 2018) can spur a virtuous cycle of predictions, experiments, and model revision that will bring us to a richer understanding of how the brain uses episodic memory in naturalistic situations.

Acknowledgements

This work was supported by a Multi-University Research Initiative grant awarded to KAN and UH (ONR/DoD N00014-17-1-2961). We are grateful for the feedback we have received from

members of the Princeton Computational Memory Lab, the Hasson Lab, and the labs of our MURI collaborators Charan Ranganath, Lucia Melloni, Jeffrey Zacks, and Samuel Gershman.

Code

Github repo: https://github.com/qihongl/learn-hippo

Appendices

Methods detail - Inter-subject correlation

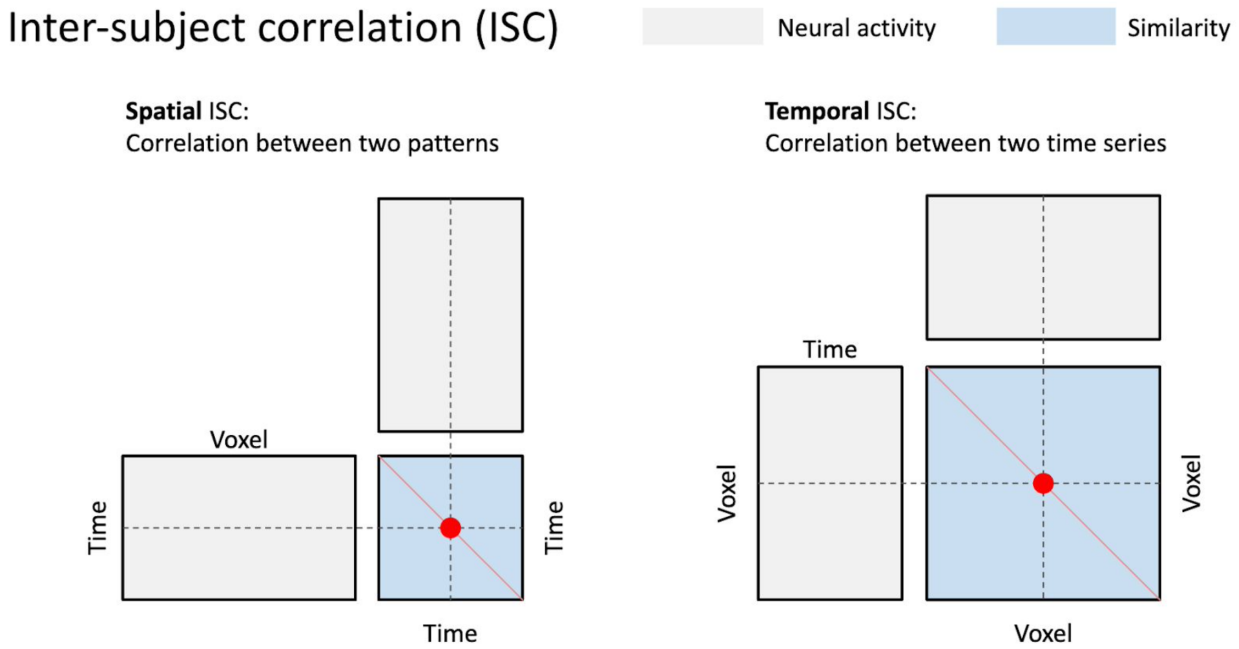

Figure 30

Inter-subject correlation. Inter-subject correlation (ISC) (Hasson et al., 2004) measures the similarity of neural representations across subjects. The spatial ISC at time t is the similarity of spatial patterns between subject i and subject j at time t. The temporal ISC for the k-th voxel/unit is the similarity of the activity time courses between subject i and subject j for the k-th voxel/unit. The red diagonal lines in the two similarity matrices correspond to spatial and temporal ISC values. ISC assumes some notion of alignment across subjects. Namely, the k-th voxel/unit between subject i and subject j should have some correspondence. Intersubject functional connectivity (ISFC) includes the off-diagonal entries.

Model parameters

Number of hidden units:

- The LSTM layer: 194
- The decision layer: 128

The level of competition: 0.8

Training:

- Supervised pre-training: 600
- Reinforcement learning: 400
- Number of trials per epoch: 256

Learning rate:

- Initial learning rate: 7e-4
- Minimal learning rate: 1e-8
- Schedule: The learning rate decays by 1/2 if the average event prediction accuracy minus mistakes stays within 0.1% from the previous best loss for 30 consecutive epochs.

Optimizer: Adam (Kingma & Ba, 2014).

Weight initialization scheme: orthogonal initialization with gain of 1 (Saxe et al., 2014).

The weight of the entropy regularization term: 0.1

Initial cell state of the LSTM: a random vector $\sim$ isotropic Gaussian(0, .1)

Parameters of the task environment

The length of the event graph / event sequences: 16

The branching factor of the event graph / the number of possible events at time t: 4

Average proportion of features observed in each event sequence: 70% for part one, 100% for part two. When a feature is withheld, the model observes a zero vector.

Penalty: In *Simulation 1*, the penalty value for the low/high condition was set to 0/4 during training and testing. In *Simulations 2 and 3*, all models were trained on a range of penalty values, sampled from 0 to 4 uniformly. Then, in the low/high penalty condition, we tested the same set of models with penalty of 0/4. In *Simulation 4* through *Simulation 8*, all models were trained on a range of penalty values, sampled from 0 to 4 uniformly, and tested on a penalty value of 2, the mean of training distribution. In *Simulation 9*, some models were trained and tested with the penalty value set to 0; other models were trained and tested with the penalty value set to 2; and still other models were trained and tested with the penalty value set to 4.
The level of event-similarity: For all simulations except for *Simulation 3*, we sampled event sequences from a fixed event schema while making sure different event sequences in the same trial were not identical – the proportion of shared features was constrained to be smaller than 90%.

Parameters specific to particular simulations

In *Simulation 3*, the level of event-similarity was constrained to be the following:

- Low: random sampling, restricting the event similarity to [0, .4]
- High: random sampling, restricting the event similarity to [.35, .9]

In *Simulation 8*, since the model got to observe three event sequences from a single situation, we lowered the average proportion of features observed in the first two blocks of each event sequence to 60%, to encourage the model to combine information across event sequences (100% of features were observed in the third and final block of each sequence). Otherwise the simulation parameters were the same as other simulations (listed in the above sections).

In *Simulation 9*, half (8) of the time points had prototypical events, meaning that one of the events at that time point happened with higher probability. We operationalized schema strength as the probability of the prototypical (higher-probability) event. We compared models trained with schema strength from 0.25 (the control condition, the same as other simulations with no prototypical events) to 0.95. Note that when the schema level was higher, the average similarity across events was higher as well. Otherwise the simulation parameters were the same as other simulations (listed in the above sections).

References

Addis, D. R. (2018). Are episodic memories special? on the sameness of remembered and imagined event simulation. *Journal of the Royal Society of New Zealand*, *48*(2-3), 64–88.

Addis, D. R., Pan, L., Vu, M.-A., Laiser, N., & Schacter, D. L. (2009). Constructive episodic simulation of the future and the past: Distinct subsystems of a core brain network mediate imagining and remembering. *Neuropsychologia*, *47*(11), 2222–2238.

Addis, D. R., Wong, A. T., & Schacter, D. L. (2007). Remembering the past and imagining the future: Common and distinct neural substrates during event construction and elaboration. *Neuropsychologia*, *45*(7), 1363–1377.

Alba, J. W., & Hasher, L. (1983). Is memory schematic? *Psychological Bulletin*, *93*(2), 203–231.

Aldous, D. J. (1985). Exchangeability and related topics (D. J. Aldous, I. A. Ibragimov, & J. Jacod, Eds.). In D. J. Aldous, I. A. Ibragimov, & J. Jacod (Eds.), *École d'été de probabilités de Saint-Flour XIII — 1983*, Springer Berlin Heidelberg.

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, *128*(2), 186–197.

Antony, J. W., Hartshorne, T. H., Pomeroy, K., Gureckis, T. M., Hasson, U., McDougle, S. D., & Norman, K. A. (2021). Behavioral, physiological, and neural signatures of surprise during naturalistic sports viewing. *Neuron*, *109*, 1–14.

Backus, A. R., Schoffelen, J.-M., Szebényi, S., Hanslmayr, S., & Doeller, C. F. (2016). Hippocampal-Prefrontal theta oscillations support memory integration. *Current Biology*, *26*(4), 450–457.

Baddeley, A. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences*, *4*(11), 417–423.

Bakkour, A., Palombo, D. J., Zylberberg, A., Kang, Y. H., Reid, A., Verfaellie, M., Shadlen, M. N., & Shohamy, D. (2019). The hippocampus supports deliberation during value-based decisions. *eLife*, *8*.

Baldassano, C., Chen, J., Zadbood, A., Pillow, J. W., Hasson, U., & Norman, K. A. (2017). Discovering event structure in continuous narrative perception and memory. *Neuron*, *95*(3), 709–721.e5.

Barnett, A. J., Reilly, W., Dimsdale-Zucker, H., Mizrak, E., Reagh, Z., & Ranganath, C. (2020). Organization of cortico-hippocampal networks in the human brain. *bioRxiv*, 2020.06.09.142166.

Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology*. New York, NY, US, Cambridge University Press.

Beaty, R. E., Thakral, P. P., Madore, K. P., Benedek, M., & Schacter, D. L. (2018). Core network contributions to remembering the past, imagining the future, and thinking creatively. *Journal of Cognitive Neuroscience*, *30*(12), 1939–1951.

Bein, O., Duncan, K., & Davachi, L. (2020). Mnemonic prediction errors bias hippocampal states. *Nature Communications*, *11*(1), 3451.

Ben-Yakov, A., & Dudai, Y. (2011). Constructing realistic engrams: Poststimulus activity of hippocampus and dorsal striatum predicts subsequent episodic memory. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *31*(24), 9032–9042.

Ben-Yakov, A., Eshel, N., & Dudai, Y. (2013). Hippocampal immediate poststimulus activity in the encoding of consecutive naturalistic episodes. *Journal of Experimental Psychology. General*, *142*(4), 1255–1263.

Ben-Yakov, A., & Henson, R. N. (2018). The hippocampal film editor: Sensitivity and specificity to event boundaries in continuous experience. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *38*(47), 10057–10068.

Ben-Yakov, A., Rubinson, M., & Dudai, Y. (2014). Shifting gears in hippocampus: Temporal dissociation between familiarity and novelty signatures in a single event. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *34*(39), 12973–12981.

Biderman, N., Bakkour, A., & Shohamy, D. (2020). What are memories for? the hippocampus bridges past experience with future decisions. *Trends in Cognitive Sciences*, *0*(0).

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, *113*(4), 700–765.

Bonasia, K., Sekeres, M. J., Gilboa, A., Grady, C. L., Winocur, G., & Moscovitch, M. (2018). Prior knowledge modulates the neural substrates of encoding and retrieving naturalistic events at short and long delays. *Neurobiology of Learning and Memory*, *153*(Pt A), 26–39.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, *23*(5), 408–422.

Brewer, W. F., & Treyens, J. C. (1981). Role of schemata in memory for places. *Cognitive Psychology*, *13*(2), 207–230.

Brunec, I. K., Robin, J., Olsen, R. K., Moscovitch, M., & Barense, M. D. (2020). Integration and differentiation of hippocampal memory traces. *Neuroscience and Biobehavioral Reviews*, *118*, 196–208.

Chanales, A. J. H., Oza, A., Favila, S. E., & Kuhl, B. A. (2017). Overlap among spatial memories triggers repulsion of hippocampal representations. *Current Biology*, *27*(15), 2307–2317.e5.

Chang, C. H. C., Lazaridi, C., Yeshurun, Y., Norman, K. A., & Hasson, U. (2020). Relating the past with the present: Information integration and segregation during ongoing narrative processing. *bioRxiv*.

Chen, C., Lu, Q., Beukers, A., Baldassano, C., & Norman, K. A. (2019). Learning to perform Role-Filler binding with schematic knowledge. *arXiv*, 1902.09006.

Chen, J., Honey, C. J., Simony, E., Arcaro, M. J., Norman, K. A., & Hasson, U. (2016). Accessing Real-Life episodic information from minutes versus hours earlier modulates hippocampal and High-Order cortical dynamics. *Cerebral Cortex*, *26*(8), 3428–3441.

Chen, P.-H., Chen, J., Yeshurun, Y., Hasson, U., Haxby, J., & Ramadge, P. J. (2015). A Reduced-Dimension fMRI shared response model. *Advances in Neural Information Processing Systems 28*.

Chien, H.-Y. S., & Honey, C. J. (2020). Constructing and forgetting temporal context in the human cerebral cortex. *Neuron*, *106*(4), 675–686.e11.

Clewett, D., DuBrow, S., & Davachi, L. (2019). Transcending time in the brain: How event memories are constructed from experience. *Hippocampus*, *29*(3), 162–183.

Cohn-Sheehy, B. I., Delarazan, A. I., Reagh, Z. M., Crivelli-Decker, J. E., Zacks, J. M., & Ranganath, C. (2020). Bridges, not walls: The hippocampus builds narrative memories across distant events. *bioRxiv*.

Collins, A. G. E. (2019). Reinforcement learning: Bringing together computation and cognition. *Current Opinion in Behavioral Sciences*, *29*, 63–68.

Cooper, R. A., Kurkela, K. A., Davis, S. W., & Ritchey, M. (2020). Mapping the organization and dynamics of the posterior medial network during movie watching. *bioRxiv*, 2020.10.21.348953.

Corkin, S. (2002). What's new with the amnesic patient H.M.? *Nature Reviews Neuroscience*, *3*(2), 153–160.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*.

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, *69*(6), 1204–1215.

De Brigard, F., Addis, D. R., Ford, J. H., Schacter, D. L., & Giovanello, K. S. (2013). Remembering what could have happened: Neural correlates of episodic counterfactual thinking. *Neuropsychologia*, *51*(12), 2401–2414.

Dhamija, A. R., Günther, M., & Boult, T. (2018). Reducing network agnostophobia, 9157–9168.

DuBrow, S., & Davachi, L. (2014). Temporal memory is shaped by encoding stability and intervening item reactivation. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *34*(42), 13998–14005.

DuBrow, S., & Davachi, L. (2016). Temporal binding within and across events. *Neurobiology of Learning and Memory*, *134 Pt A*, 107–114.

Dudai, Y. (2009). Predicting not to predict too much: How the cellular machinery of memory anticipates the uncertain future. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *364*(1521), 1255–1262.

Dudai, Y., & Eisenberg, M. (2004). Rites of passage of the engram: Reconsolidation and the lingering consolidation hypothesis. *Neuron*, *44*(1), 93–100.

Duncan, K., Sadanand, A., & Davachi, L. (2012). Memory's penumbra: Episodic memory decisions induce lingering mnemonic biases. *Science*, *337*(6093), 485–487.

Duncan, K., & Schlichting, M. (2018). Hippocampal representations as a function of time, subregion, and brain state. *Neurobiology of Learning and Memory*, *153*(Pt A), 40–56.

Duncan, K., Semmler, A., & Shohamy, D. (2019). Modulating the use of multiple memory systems in value-based decisions with contextual novelty. *Journal of Cognitive Neuroscience*, *31*(10), 1455–1467.

Duncan, K., & Shohamy, D. (2016). Memory states influence value-based decisions. *Journal of Experimental Psychology: General*, *145*(11), 1420–1426.

Elman, J. L., & McRae, K. (2019). A model of event knowledge. *Psychological Review*, *126*(2), 252–291.

Ezzyat, Y., & Davachi, L. (2011). What constitutes an episode in episodic memory? *Psychological Science*, *22*(2), 243–252.

Ezzyat, Y., & Davachi, L. (2014). Similarity breeds proximity: Pattern similarity within and across contexts is related to later mnemonic judgments of temporal proximity. *Neuron*, *81*(5), 1179–1189.

Favila, S. E., Chanales, A. J. H., & Kuhl, B. A. (2016). Experience-dependent hippocampal pattern differentiation prevents interference during subsequent learning. *Nature Communications*, *7*, 11066.

Franklin, N. T., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2020). Structured event memory: A neuro-symbolic model of event cognition. *Psychological Review*, *127*(3), 327–361.

Gershman, S. J., Blei, D. M., & Niv, Y. (2010). Context, learning, and extinction. *Psychological Review*, *117*(1), 197–209.

Gershman, S. J., Norman, K. A., & Niv, Y. (2015). Discovering latent causes in reinforcement learning. *Current Opinion in Behavioral Sciences*, *5*, 43–50.

Gilboa, A., & Marlatte, H. (2017). Neurobiology of schemas and Schema-Mediated memory. *Trends in Cognitive Sciences*, *21*(8), 618–631.

Gilboa, A., Winocur, G., Rosenbaum, R. S., Poreh, A., Gao, F., Black, S. E., Westmacott, R., & Moscovitch, M. (2006). Hippocampal contributions to recollection in retrograde and anterograde amnesia. *Hippocampus*, *16*(11), 966–980.

Grandvalet, Y., & Bengio, Y. (2006). Entropy regularization (Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, Ed.). In Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien (Ed.), *Semi-Supervised Learning*.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv*, 1410.5401.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., Badia, A. P., Hermann, K. M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., & Hassabis, D. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471–476.

Greve, A., Cooper, E., Kaula, A., Anderson, M. C., & Henson, R. (2017). Does prediction error drive one-shot declarative learning? *Journal of Memory and Language*, *94*, 149–165.

Griffiths, B. J., & Fuentemilla, L. (2020). Event conjunction: How the hippocampus integrates episodic memories across event boundaries. *Hippocampus*, *30*(2), 162–171.

Hardt, O., Einarsson, E. Ö., & Nader, K. (2010). A bridge over troubled water: Reconsolidation as a link between cognitive and neuroscientific memory research traditions. *Annual Review of Psychology*, *61*(1), 141–167.

Hassabis, D., Kumaran, D., Vann, S. D., & Maguire, E. A. (2007). Patients with hippocampal amnesia cannot imagine new experiences. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(5), 1726–1731.

Hasselmo, M. E., & Wyble, B. P. (1997). Free recall and recognition in a network model of the hippocampus: Simulating effects of scopolamine on human memory function. *Behavioural Brain Research*, *89*(1-2), 1–34.

Hasson, U., Chen, J., & Honey, C. J. (2015). Hierarchical process memory: Memory as an integral component of information processing. *Trends in Cognitive Sciences*, *19*(6), 304–313.

Hasson, U., Nir, Y., Levy, I., Fuhrmann, G., & Malach, R. (2004). Intersubject synchronization of cortical activity during natural vision. *Science*, *303*(5664), 1634–1640.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, *293*(5539), 2425–2430.

Haxby, J. V., Guntupalli, J. S., Connolly, A. C., Halchenko, Y. O., Conroy, B. R., Gobbini, M. I., Hanke, M., & Ramadge, P. J. (2011). A common, high-dimensional model of the representational space in human ventral temporal cortex. *Neuron*, *72*(2), 404–416.

Haxby, J. V., Guntupalli, J. S., Nastase, S. A., & Feilong, M. (2020). Hyperalignment: Modeling shared information encoded in idiosyncratic cortical topographies. *eLife*, *9*, e56601.

Henson, R. N., & Gagnepain, P. (2010). Predictive, interactive multiple memory systems. *Hippocampus*, *20*(11), 1315–1326.

Himberger, K. D., Chien, H.-Y., & Honey, C. J. (2018). Principles of temporal processing across the cortical hierarchy. *Neuroscience*, *389*, 161–174.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Holdstock, J. S., Mayes, A. R., Roberts, N., Cezayirli, E., Isaac, C. L., O'Reilly, R. C., & Norman, K. A. (2002). Under what conditions is recognition spared relative to recall after selective hippocampal damage in humans? *Hippocampus*, *12*(3), 341–351.

Holdstock, J. S., Shaw, C., & Aggleton, J. P. (1995). The performance of amnesic subjects on tests of delayed matching-to-sample and delayed matching-to-position. *Neuropsychologia*, *33*(12), 1583–1596.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(8), 2554–2558.

Hulbert, J. C., & Norman, K. A. (2015). Neural differentiation tracks improved recall of competing memories following interleaved study and retrieval practice. *Cerebral Cortex*, *25*(10), 3994–4008.

Kahana, M. J. (2012). *Foundations of human memory*. Oxford University Press, USA.

Ketz, N., Morkonda, S. G., & O'Reilly, R. C. (2013). Theta coordinated error-driven learning in the hippocampus. *PLOS Computational Biology*, *9*(6), e1003067.

Kim, G., Norman, K. A., & Turk-Browne, N. B. (2017). Neural differentiation of incorrectly predicted memories. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *37*(8), 2022–2031.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*, 1412.6980.

Kumar, M., Ellis, C. T., Lu, Q., Zhang, H., Capotă, M., Willke, T. L., Ramadge, P. J., Turk-Browne, N. B., & Norman, K. A. (2020). BrainIAK tutorials: User-friendly learning materials for advanced fMRI analysis. *PLoS computational biology*, *16*(1), e1007549.

Kumar, M., Michael Anderson, Antony, J., Baldassano, C., Brooks, P. P., Cai, M. B., Chen, P.-H. C., Ellis, C. T., Henselman-Petrusek, G., Huberdeau, D., Hutchinson, J. B., Li, P. Y., Lu, Q., Manning, J. R., Mennen, A. C., Nastase, S. A., Richard, H., Schapiro, A. C., Schuck, N. W., ... Norman, K. A. (2020). BrainIAK: The brain imaging analysis kit. *OSF Preprints*.

Lerner, Y., Honey, C. J., Silbert, L. J., & Hasson, U. (2011). Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *31*(8), 2906–2915.

Lewis-Peacock, J. A., & Norman, K. A. (2014). Competition between items in working memory leads to forgetting. *Nature Communications*, *5*, 5768.

Li, Y., Yosinski, J., Clune, J., Lipson, H., & Hopcroft, J. (2015). Convergent learning: Do different neural networks learn the same representations? *Proceedings of Machine Learning Research*, *44*, 196–212.

Lu, Q., Chen, P.-H., Pillow, J. W., Ramadge, P. J., Norman, K. A., & Hasson, U. (2018). Shared representational geometry across neural networks. *arXiv*, 1811.11684.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419–457.

Meng, Q., Chen, W., Zheng, S., Ye, Q., & Liu, T.-Y. (2018). Optimizing neural networks in the equivalent class space. *arXiv*, 1802.03713.

Michelmann, S., Price, A. R., Aubrey, B., Doyle, W. K., Friedman, D., Dugan, P., Devinsky, O., Devore, S., Flinker, A., Hasson, U., & Norman, K. A. (2020). Moment-by-moment tracking of naturalistic learning and its underlying hippocampo-cortical interactions. *bioRxiv*.

Milner, B., Corkin, S., & Teuber, H.-L. (1968). Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of H.M. *Neuropsychologia*, *6*(3), 215–234.

Misra, D., Langford, J., & Artzi, Y. (2017). Mapping instructions and visual observations to actions with reinforcement learning. *arXiv*, 1704.08795.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *arXiv*, 1602.01783.

Morris, R. (1984). Developments of a water-maze procedure for studying spatial learning in the rat. *Journal of Neuroscience Methods*, *11*(1), 47–60.

Nagabandi, A., Kahn, G., Fearing, R. S., & Levine, S. (2017). Neural network dynamics for Model-Based deep reinforcement learning with Model-Free Fine-Tuning. *arXiv*, 1708.02596.

Nastase, S. A., Gazzola, V., Hasson, U., & Keysers, C. (2019). Measuring shared responses across subjects using intersubject correlation. *Social Cognitive and Affective Neuroscience*, *14*(6), 667–685.

Nastase, S. A., Goldstein, A., & Hasson, U. (2020). Keep it real: Rethinking the primacy of experimental control in cognitive neuroscience. *NeuroImage*, *222*, 117254.

Norman, K. A. (2010). How hippocampus and cortex contribute to recognition memory: Revisiting the complementary learning systems model. *Hippocampus*, *20*(11), 1217–1227.

Norman, K. A., Detre, G., & Polyn, S. M. (2008). Computational models of episodic memory. In R. Sun (Ed.), *The Cambridge handbook of computational psychology* (pp. 189–225). Cambridge University Press.

Norman, K. A., & O'Reilly, R. C. (2003). Modeling hippocampal and neocortical contributions to recognition memory: A complementary-learning-systems approach. *Psychological Review*, *110*(4), 611–646.

Norman, K. A., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: Multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, *10*(9), 424–430.

Oedekoven, C. S. H., Keidel, J. L., Anderson, S., Nisbet, A., & Bird, C. M. (2019). Effects of amnesia on processing in the hippocampus and default mode network during a naturalistic memory task: A case study. *Neuropsychologia*, *132*, 107104.

O'Reilly, R. C., & McClelland, J. L. (1994). Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off. *Hippocampus*, *4*(6), 661–682.

Palombo, D. J., Hayes, S. M., Reid, A. G., & Verfaellie, M. (2019). Hippocampal contributions to value-based learning: Converging evidence from fMRI and amnesia. *Cognitive, Affective & Behavioral Neuroscience*, *19*(3), 523–536.

Palombo, D. J., Keane, M. M., & Verfaellie, M. (2015). How does the hippocampus shape decisions? *Neurobiology of Learning and Memory*, *125*, 93–97.

Patil, A., & Duncan, K. (2018). Lingering cognitive states shape fundamental mnemonic abilities. *Psychological Science*, *29*(1), 45–55.

Pitman, J. (2006). *Combinatorial stochastic processes: Ecole d'eté de probabilités de Saint-Flour XXXII – 2002* (J. Picard, Ed.). Springer, Berlin, Heidelberg.

Polyn, S. M., Norman, K. A., & Kahana, M. J. (2009). A context maintenance and retrieval model of organizational processes in free recall. *Psychological Review*, *116*(1), 129–156.

Preston, A. R., & Eichenbaum, H. (2013). Interplay of hippocampus and prefrontal cortex in memory. *Current Biology*, *23*(17), R764–73.

Pritzel, A., Uria, B., Srinivasan, S., Badia, A. P., Vinyals, O., Hassabis, D., Wierstra, D., & Blundell, C. (2017). Neural episodic control (D. Precup & Y. W. Teh, Eds.). *Proceedings of Machine Learning Research*, *70*, 2827–2836.

Radvansky, G. A., Krawietz, S. A., & Tamplin, A. K. (2011). Walking through doorways causes forgetting: Further explorations. *Quarterly Journal of Experimental Psychology*, *64*(8), 1632–1645.

Ranganath, C., & Rainer, G. (2003). Neural mechanisms for detecting and remembering novel events. *Nature Reviews Neuroscience*, *4*(3), 193–202.

Ranganath, C., & Ritchey, M. (2012). Two cortical systems for memory-guided behaviour. *Nature Reviews Neuroscience*, *13*(10), 713–726.

Reagh, Z. M., Delarazan, A. I., Garber, A., & Ranganath, C. (2020). Aging alters neural activity at event boundaries in the hippocampus and posterior medial network. *Nature Communications*, *11*(1), 3980.

Richmond, L. L., & Zacks, J. M. (2017). Constructing experience: Event models from perception to action. *Trends in Cognitive Sciences*, *21*(12), 962–980.

Ritchey, M., & Cooper, R. A. (2020). Deconstructing the posterior medial episodic network. *Trends in Cognitive Sciences*, *24*(6), 451–465.

Ritter, S. (2019). *Meta-reinforcement learning with episodic recall: An integrative theory of reward-driven learning* (Doctoral dissertation). Princeton University.

Ritter, S., Wang, J. X., Kurth-Nelson, Z., Jayakumar, S. M., Blundell, C., Pascanu, R., & Botvinick, M. (2018). Been there, done that: Meta-Learning with episodic recall. *Proceedings of the International Conference on Machine Learning*.

Ritvo, V. J. H., Turk-Browne, N. B., & Norman, K. A. (2019). Nonmonotonic plasticity: How memory retrieval drives learning. *Trends in Cognitive Sciences*, *23*(9), 726–742.

Rolls, E. T. (2010). Attractor networks. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(1), 119–134.

Rolls, E. T. (2013). The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in Systems Neuroscience*, *7*, 74.

Rouhani, N., Norman, K. A., & Niv, Y. (2018). Dissociable effects of surprising rewards on learning and memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *44*(9), 1430–1443.

Rouhani, N., Norman, K. A., Niv, Y., & Bornstein, A. M. (2020). Reward prediction errors create event boundaries in memory. *Cognition*, *203*, 104269.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *International Conference on Learning Representations*.

Schapiro, A. C., Kustner, L. V., & Turk-Browne, N. B. (2012). Shaping of object representations in the human medial temporal lobe based on temporal regularities. *Current Biology*, *22*(17), 1622–1627.

Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M., & Norman, K. A. (2017). Complementary learning systems within the hippocampus: A neural network modelling approach to reconciling episodic memory with statistical learning. *Philosophical Transactions of the Royal Society of London. Series B, Biological sciences*, *372*(1711), 20160049.

Schapiro, A. C., Turk-Browne, N. B., Norman, K. A., & Botvinick, M. M. (2016). Statistical learning of temporal community structure in the hippocampus. *Hippocampus*, *26*(1), 3–8.

Schlichting, M. L., Mumford, J. A., & Preston, A. R. (2015). Learning-related representational changes reveal dissociable integration and separation signatures in the hippocampus and prefrontal cortex. *Nature Communications*, *6*, 8151.

Schlichting, M. L., & Preston, A. R. (2015). Memory integration: Neural mechanisms and implications for behavior. *Current Opinion in Behavioral Sciences*, *1*, 1–8.

Schlichting, M. L., & Preston, A. R. (2016). Hippocampal-medial prefrontal circuit supports memory updating during learning and post-encoding rest. *Neurobiology of Learning and Memory*, *134 Pt A*, 91–106.

Schlichting, M. L., Zeithamova, D., & Preston, A. R. (2014). CA1 subfield contributions to memory integration and inference. *Hippocampus*, *24*(10), 1248–1260.

Sederberg, P. B., Howard, M. W., & Kahana, M. J. (2008). A context-based theory of recency and contiguity in free recall. *Psychological Review*, *115*(4), 893–912.

Sherman, B. E., & Turk-Browne, N. B. (2020). Statistical prediction of the future impairs episodic encoding of the present. *Proceedings of the National Academy of Sciences of the United States of America*, *117*(37), 22760–22770.

Shohamy, D., & Turk-Browne, N. B. (2013). Mechanisms for widespread hippocampal involvement in cognition. *Journal of Experimental Psychology: General*, *142*(4), 1159–1170.

Shohamy, D., & Wagner, A. D. (2008). Integrating memories in the human brain: Hippocampal-midbrain encoding of overlapping events. *Neuron*, *60*(2), 378–389.

Simony, E., Honey, C. J., Chen, J., Lositsky, O., Yeshurun, Y., Wiesel, A., & Hasson, U. (2016). Dynamic reconfiguration of the default mode network during narrative comprehension. *Nature Communications*, *7*, 12141.

Sonkusare, S., Breakspear, M., & Guo, C. (2019). Naturalistic stimuli in neuroscience: Critically acclaimed. *Trends in Cognitive Sciences*, *23*(8), 699–714.

Stachenfeld, K. L., Botvinick, M. M., & Gershman, S. J. (2017). The hippocampus as a predictive map. *Nature Neuroscience*, *20*(11), 1643–1653.

Stawarczyk, D., Bezdek, M. A., & Zacks, J. M. (2019). Event representations and predictive processing: The role of the midline default network core. *Topics in Cognitive Science*, *30*, 1345.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Takahashi, Y., Schoenbaum, G., & Niv, Y. (2008). Silencing the critics: Understanding the effects of cocaine sensitization on dorsolateral and ventral striatum in the context of an actor/critic model. *Frontiers in Neuroscience*, *2*(1), 86–99.

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review*, *108*(3), 550–592.

van Kesteren, M. T. R., Ruiter, D. J., Fernández, G., & Henson, R. N. (2012). How schema and novelty augment memory formation. *Trends in Neurosciences*, *35*(4), 211–219.

Vikbladh, O. M., Meager, M. R., King, J., Blackmon, K., Devinsky, O., Shohamy, D., Burgess, N., & Daw, N. D. (2019). Hippocampal contributions to Model-Based planning and spatial memory. *Neuron*, *102*(3), 683–693.e4.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, *21*(6), 860–868.

Wang, S.-H., & Morris, R. G. M. (2010). Hippocampal-neocortical interactions in memory formation, consolidation, and reconsolidation. *Annual Review of Psychology*, *61*, 49–79, C1–4.

Wayne, G., Hung, C.-C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J., Mirowski, P., Leibo, J. Z., Santoro, A., Gemici, M., Reynolds, M., Harley, T., Abramson, J., Mohamed, S., Rezende, D., Saxton, D., Cain, A., Hillier, C., ... Lillicrap, T. (2018). Unsupervised predictive memory in a Goal-Directed agent. *arXiv*, 1803.10760.

Whittington, J. C. R., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., & Behrens, T. E. J. (2020). The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, *183*(5), 1249–1263.e23.

Yassa, M. A., & Stark, C. E. L. (2011). Pattern separation in the hippocampus. *Trends in Neurosciences*, *34*(10), 515–525.

Yonelinas, A. P. (2002). The nature of recollection and familiarity: A review of 30 years of research. *Journal of Memory and Language*, *46*(3), 441–517.

Zacks, J. M. (2020). Event perception and memory. *Annual Review of Psychology*, *71*, 165–191.

Zacks, J. M., Kurby, C. A., Eisenberg, M. L., & Haroutunian, N. (2011). Prediction error associated with the perceptual segmentation of naturalistic events. *Journal of Cognitive Neuroscience*, *23*(12), 4057–4066.

Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., & Reynolds, J. R. (2007). Event perception: A mind-brain perspective. *Psychological Bulletin*, *133*(2), 273–293.

Zeithamova, D., Dominick, A. L., & Preston, A. R. (2012). Hippocampal and ventral medial prefrontal activation during retrieval-mediated learning supports novel inference. *Neuron*, *75*(1), 168–179.

Zeithamova, D., & Preston, A. R. (2010). Flexible memories: Differential roles for medial temporal lobe and prefrontal cortex in cross-episode binding. *The Journal of Neuroscience: the official journal of the Society for Neuroscience*, *30*(44), 14676–14684.

Zeithamova, D., Schlichting, M. L., & Preston, A. R. (2012). The hippocampus and inferential reasoning: Building memories to navigate future decisions. *Frontiers in Human Neuroscience*, *6*, 70.

Zuo, X., Honey, C. J., Barense, M. D., Crombie, D., Norman, K. A., Hasson, U., & Chen, J. (2020). Temporal integration of narrative information in a hippocampal amnesic patient. *NeuroImage*, *213*(116658).