Consolidation of sequential experience into a deep generative network explains human memory, prediction and planning

Eleanor Spens
Institute of Cognitive Neuroscience
University College London
eleanor.spens.20@ucl.ac.uk

Neil Burgess
Institute of Cognitive Neuroscience
University College London
n.burgess@ucl.ac.uk

Abstract

The consolidation of sequential experience is thought to enable efficient schema-based reconstruction of the past and prediction of the future, but the mechanism is unknown. Here, we present a computational model in which sequences are rapidly encoded in the hippocampus and replayed to train a neocortical deep generative network to predict the next item in each sequence. This is simulated using generative pre-trained transformers, a variety of large language model. As well as capturing the gist of specific episodes, the neocortical network extracts statistical patterns that generalise to new situations. This model explains human performance on statistical learning and structural inference tasks, and accounts for gist or schema-based distortions in memories of narratives. It also shows how recent memory can contribute to inference and planning, with specific memories retrieved from the hippocampus providing the context in working memory for prediction using the 'general knowledge' of the neocortical network; we model this hippocampal and neocortical interaction as 'retrieval-augmented generation'. Furthermore, we show how hippocampal traces could be compressed by combining concepts and details, with the conceptual representation optimised for efficient reconstruction. The model suggests how episodic, semantic and working memory interact in the consolidation, (re)construction and planning of sequential experience.

1 Introduction

There is a long history of examining how different types of memory interact to support adaptive behaviour by using conceptual or computational models. A key question is how short, medium, and long-term stores support encoding, consolidation, and recall across the 'lifespan' of a memory (e.g. Atkinson and Shiffrin, 1968). Crucially, semantic memory is thought to be extracted from episodic memory through consolidation (Marr, 1970, 1971; McClelland et al., 1995), learning statistical relationships from experience to aid prediction and thus survival (Schacter et al., 2007). Both semantic and episodic memory can interact within working memory when reconstructing what happened in the past, or imagining or planning what will happen in the future, with episodic memory contributing specifics and semantic memory contributing generalisable knowledge (Baddeley, 2001). The conception of memory as (re)constructed rather than veridical (Bartlett, 1932) highlights the

potential of recently developed deep generative networks to shed new light on these questions. Here we use generative pre-trained transformers (GPTs; Radford et al., 2019) to model how neocortical networks combine with the hippocampus to extract semantic knowledge from sequential experiences, and to support remembering, inference and sequential planning.

The idea that memories are rapidly encoded by the hippocampus and then replayed over the course of systems consolidation to train a generative or predictive model of the world in neocortex has received much recent interest. The combined system is thought to support multiple cognitive functions including episodic memory, semantic memory, imagination, and inference (Fayyaz et al., 2022; Káli & Dayan, 2000, 2002; Spens & Burgess, 2024). This provides a mechanistic account of the theory that episodic memories are reconstructions that are influenced by our beliefs, i.e. that recall involves 'predicting' the past (Hemmer & Steyvers, 2009; Nagy et al., 2020, 2025). However previous work has focused on static patterns, whereas episodes are sequential, and arguably the most crucial function of semantic knowledge involves prediction of what will come next.

Memory helps to predict the future, not just recall the past (Schacter et al., 2007). How do sequential memories inform future behaviour through inference and planning, and how does this change over the lifespan of a memory? Planning often requires anticipating the outcome of a possible action, e.g. the sequence of future states and rewards that would follow it. The mental simulation of future states to aid decision-making is known as 'model-based' planning (Daw et al., 2011; Doll et al., 2015), which improves with consolidation (Vikbladh et al., 2024). We suggest that the generative network trained on memories can simulate events to support behaviour, such as when deciding between possible actions, or when inferring the next state given the sequence so far. (See also Chen et al., 2021.)

Machine learning has begun to show how deep neural networks can act as task-general 'foundation models' (Bommasani et al., 2021). In particular, large language models (LLMs) demonstrate that complex sequential behaviours can develop as a byproduct of a simple 'next item prediction' task (Brown et al., 2020; Radford et al., 2019). Here we suggest how memory consolidation could contribute to the development of neural 'foundation models' in a similar way, with neocortex learning to reconstruct replayed memories via schemas (see Spens and Burgess, 2024). Interestingly, LLMs can memorise specific sequences as well as learning generalities (Carlini, Ippolito, et al., 2022), meaning that episodic and semantic information can be captured within a single network (once the former is fully consolidated), raising fundamental questions regarding the boundary between episodic and semantic systems.

We model consolidation as the training of autoregressive sequence models (simulated using GPT-2; Radford et al., 2019) on sequences which represent replayed hippocampal memories. Consolidation can thus be thought of as teacher-student learning (see Sun et al., 2023). The objective during training is simply to predict the next item in sequences from the training data. Once trained, the network can continue an input sequence, or generate a new sequence from scratch, by iteratively predicting the next item from the items so far (the 'prompt'). By representing a range of stimuli as sequences, we show that our model is capable of classic statistical learning (Durrant et al., 2011) and spatial / relational inference (Whittington et al., 2020) tasks, and learns to support model-based planning over time (as in Vikbladh et al., 2024). In addition, by 'consolidating' narratives into GPT-2 (Radford et al., 2019), we show that the model displays similar gist-based memory distortions to those observed in human memory for narratives (Bartlett, 1932; Bergman & Roediger, 1999; Raykov et al., 2023). (Note that whilst the generative networks' training data represent sequences replayed from the hippocampus, hippocampal encoding and retrieval are not simulated in these results.)

General knowledge and *specific* memory traces can play complementary roles in problem-solving (see Lengyel and Dayan, 2007). For example, if one meets person A, who mentions their parent B, and later that same day meets B, who mentions their parent C, inferring that A is the grandchild of C requires the hippocampal traces from the two conversations to be combined with general knowledge of family relationships in working memory. That is, using recent experiences to inform behaviour before their content has been consolidated requires both specific episodic memories from the hippocampus *and* more general knowledge from the neocortex (Robin & Moscovitch, 2017; Samborska et al., 2022). There are few models of how this interaction might work.

We propose that the hippocampal 'memory bank' and neocortical generative network work together to enable problem-solving, with the hippocampus retrieving memories relevant to a new task into working memory as context for the generative network. Specifically, one can draw inspiration from retrieval-augmented generation (RAG; Lewis et al., 2020), which involves prompting LLMs with

relevant data retrieved from external memory, as a model of drawing on relevant memories to inform predictions about a new event. A typical task for RAG might be question answering based on a set of new facts which do not feature in the training data for the chosen LLM. To ask the system a question, first the most relevant items are found in the external memory (using some similarity metric). A 'prompt' is then constructed, in which the retrieved items are given to the LLM together with the question. We suggest hippocampal-neocortical interaction may resemble RAG, but with the addition of consolidation to gradually train the 'generator' on memories from the 'retriever' (Figure 1a and d). Furthermore memories in the 'retriever' can be stored in a compact conceptual form, taking inspiration from the LLM literature on context compression (Cheng et al., 2024). In the 'full' model, the conceptual representation is optimised for the efficient reconstruction of memories from their compressed form, and all recall of hippocampal traces is retrieval-augmented generation, requiring the generative network to fill in the outline of a memory with its predictions.

Here working memory refers to the temporary retention of a limited amount of information (Baddeley, 1992), which can include episodic memories retrieved from long-term memory (Baddeley, 2000). To be more precise, we use 'working memory' as shorthand for the episodic buffer, a temporary store thought to sit between the phonological loop / visuospatial sketchpad and long term memory (Baddeley, 2000), which can accommodate considerably more than the 4 or so items which can be attended to at a given moment (Baddeley, 2000; Cowan, 2001). This is the 'workspace' into which episodic memories relevant to a task are retrieved from the hippocampus, upon which the neocortical network operates. Specifically, the context (or 'prompt') provided to an LLM can be thought of as its 'working memory' (Li et al., 2022), with the limited context length of an LLM representing the limited capacity of biological working memory. Note that the neocortical model's outputs could be conditioned on perception rather than memory by the same mechanism, if the contents of working memory are new sequences rather than ones retrieved from the hippocampus.

The resemblance between LLMs and human cognition is controversial. As with most deep neural networks, aspects of their design are biologically implausible (Whittington & Bogacz, 2019), and there is debate as to whether such models can meaningfully plan and reason at all (e.g. Bender et al., 2021; Kambhampati et al., 2024). Biological learning is also far more data efficient, with many human lifetimes' worth of language used to train leading LLMs (Warstadt & Bowman, 2022). We do *not* argue that the internal workings of LLMs resemble the brain, but suggest that a neural 'foundation model' being trained to minimise prediction error on replayed sequences provides a useful model for investigating human memory and problem solving.

In summary, we present a model of how sequential memories are initially encoded in the hippocampus and then gradually consolidated into a generative model in neocortex, through prediction error minimisation on the replayed memories. This provides a mechanism for the training of neural 'foundation models' (Bommasani et al., 2021), which can both capture specific memories *and* learn more general schematic knowledge to enable problem solving. We show how episodic memories could be combined with a neocortical world model to enable problem solving based on recent experience, inspired by 'retrieval-augmented generation' (RAG) in the LLM literature (Lewis et al., 2020). That is, we model the roles of hippocampus and neocortex in memory processing as a form of RAG, but with the hippocampal 'retriever' also training the neocortical 'generator', and the neocortical 'generator' compressing memories before they are stored in the hippocampal 'retriever'.

2 Methods

We first train sequence models (GPT-2; Radford et al., 2019) on a variety of sequence types, representing experiences encoded in the hippocampus and replayed during rest, and explore how they might support memory, statistical learning, structural inference, and model-based planning abilities. Using these models, we then simulate the interaction between neocortical and hippocampal memory systems as retrieval-augmented generation (Lewis et al., 2020). Building on this, we demonstrate how memories could be encoded in a highly compressed form in the hippocampus (Cheng et al., 2024), and explore the effect of consolidating these already abstracted events.

The training data for the generative networks is intended to represent replayed sequences from the hippocampus, although this is not modelled in Sections 3.1 and 3.2. We envisage the hippocampal network as a sequential (i.e. asymmetric) variant of the modern Hopfield network (Millidge et al., 2022; Ramsauer et al., 2020), but do not simulate errors that could occur during encoding or retrieval,

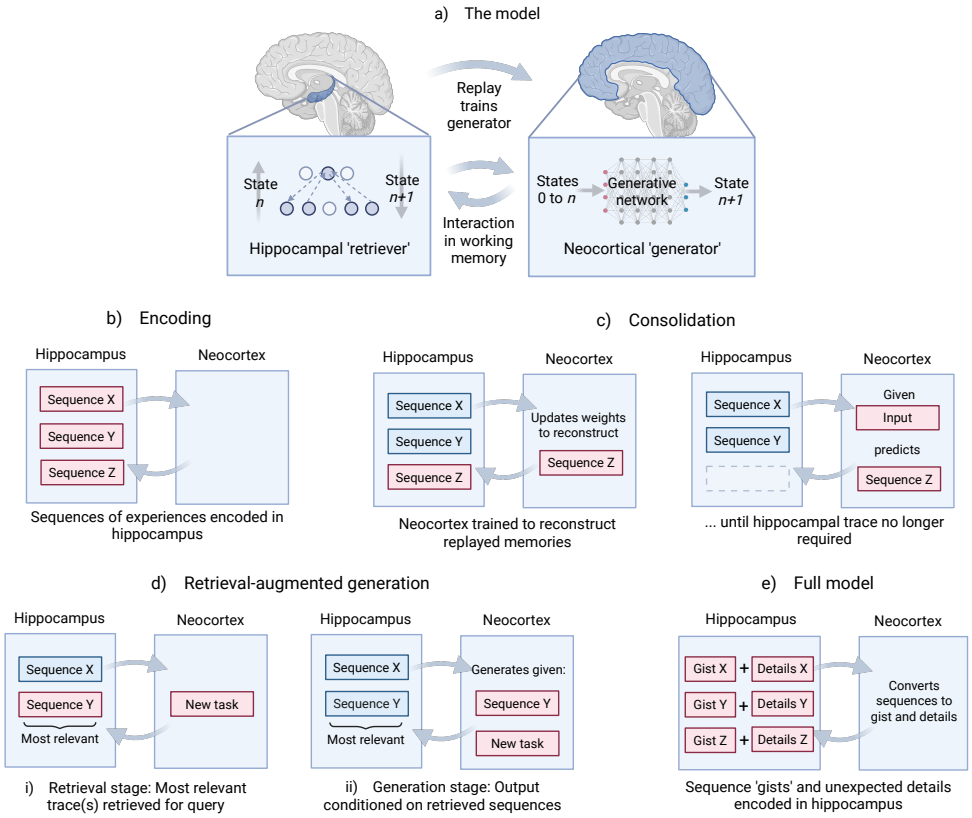

Figure 1: Summary of model. a) Episodic memories are first encoded in the hippocampus, then replayed during rest to train a generative network (simulated by an LLM). Semantic memory is acquired as a byproduct. The hippocampus ('retriever') and neocortex ('generator') can interact to play complementary roles during problem solving, resembling an adaptive variant of 'retrieval-augmented generation' (RAG; Lewis et al., 2020) in which the 'generator' is trained by the 'retriever'. Relevant hippocampal traces are retrieved into working memory and then used to condition (or 'prompt') the neocortical network. b) Sequences are first encoded in the hippocampus. c) Memories are replayed from the hippocampus during rest, and the generative network learns to capture their statistics through prediction error minimisation. Once a sequence is consolidated, the generative network can retrieve the rest of a sequence given its start, without requiring the hippocampal trace. d) Even prior to consolidation, the neocortical and hippocampal networks can work together to solve problems via RAG. e) In the 'full' model, sequence 'gists' are encoded together with unexpected details. Specifically, a vector from which the sequence can be approximately decoded is stored for each event (Cheng et al., 2024), together with the subsequences that were most surprising given this compressed version of the event.

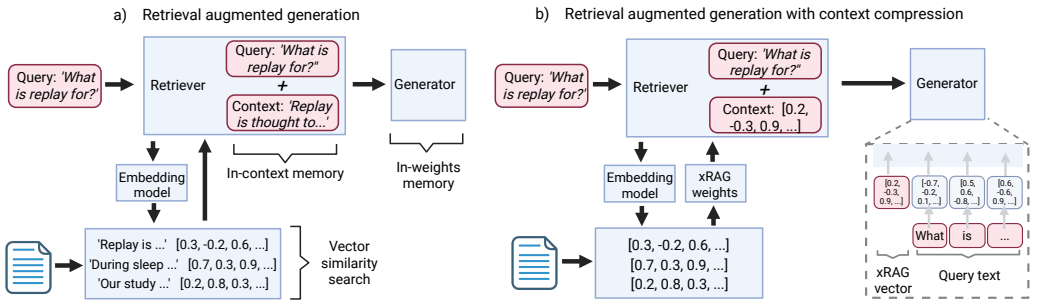

Figure 2: a) Retrieval-augmented generation (RAG; Lewis et al., 2020). When the neocortical generative network ('generator') is prompted with a query, e.g. 'What is replay for?', related information is retrieved from hippocampal memory ('retriever'). The retrieval process finds stored information with similar vectors to the query (all text is converted into vectors by a 'text embedding model'). A combined prompt is then created from the query and the retrieved information or 'context', improving the output from the generator. b) Retrieval-augmented generation with context compression using xRAG (Cheng et al., 2024). This approach trains a small network to map vectors from a text embedding model into a single token representation. This allows the context to be represented as a single token, which is fed into the first layer of an LLM together with the tokens of the query. This means the context document itself does not need to be stored in external memory; storing its vector is enough.

in order to focus on the neocortical generative network. For further details see Section 6.4.3, SI. Note that sequences are repeated many times during training, reflecting offline replay rather than online learning from a single exposure.

The 'medium' sized GPT-2 model is used, which has 345 million parameters including 24 transformer blocks. In some simulations, pre-trained GPT-2 weights (from Radford et al., 2019) are used as the starting point for further training, and in others the GPT-2 architecture is trained from scratch with randomly initialised weights. The 'further training' (i.e. 'fine-tuning') option is used for the simulations involving language (Sections 3.1 and 3.1.2) and the model-based planning simulation (Section 3.2.3) because of the complexity of the stimuli. (Pre-trained as opposed to randomly initialised weights can benefit learning even for non-linguistic sequences.) The 'from scratch' option is used in the simulations of statistical learning (Section 3.2.1) and structural inference (Section 3.2.2) because the stimuli are relatively simple and only their statistics are relevant. See Table 1 for a summary of the models trained, Section 6.3 of the SI for further details of the training procedure, and Table 10 for a summary of the hyperparameters.

Next we simulate retrieval-augmented generation. In Section 3.3.1 we use the models trained in Sections 3.2.2 and 3.2.3 to represent neocortex, while in Section 3.3.2 we use Mistral-7B-Instruct-v0.2 (Jiang, Sablayrolles, et al., 2023) to explore the decomposition of events into gists and details. In addition, pre-trained weights from Cheng et al. (2024) are used to transform events into compressed xRAG vectors. (Note that xRAG weights are only compatible with a particular LLM; Mistral-7B-Instruct-v0.2 is used because the xRAG weights are trained to transform embeddings into token representations of this specific model. But in principle the context compression mechanism is compatible with any LLM.)

LLMs capture probability distributions, allowing more flexible behaviour than rote memorisation. One important variable throughout these results is the temperature, which determines the 'sharpness' of the distribution from which output tokens are selected. Sequences sampled from the probability distribution with a higher temperature are more 'imaginative', as this adds more randomness into the next token predictions. Conversely, at lower temperatures the most probable tokens are more likely to be sampled, which increases the chance of generating memorised sequences from the training data. In contrast to sampling, 'greedy decoding' refers to simply selecting the highest probability next token. See Section 6.3.2, SI, for further details.

Simulation	Section	Training method	Model	Training data
Gist-based distortions	3.1	Further training	GPT-2 medium	English language text from Bartlett (1932) and Wikipedia (Ziadé, 2024)
Event extension and contraction	3.1.2	Further training	GPT-2 medium	English language text from ROC Stories dataset (Mostafazadeh et al., 2016)
Statistical learning	3.2.1	From scratch	GPT-2 medium	Sequences of tones from Durrant et al. (2011) represented by '2,2,3,1,...'
Structural inference (spatial)	3.2.2	From scratch	GPT-2 medium	Walks on a graph of form 'mv SOUTH sz WEST li EAST sz ...'
Structural inference (family trees)	3.2.2	From scratch	GPT-2 medium	Walks on a graph of form 'yu GRANDPARENT_OF mi SIBLING_OF vb ...'
Model-based planning	3.2.3	Further training	GPT-2 medium	Tasks from Vikbladh et al. (2024) represented by 'START: yellow vehicle, STOP: green, REWARD: animal, SEQUENCE: red animal (2), green vehicle (-1)'
Encoding gists	3.3.2	N/A (used pre-trained model)	Mistral-7B-Instruct-v0.2 plus xRAG weights	N/A (used pre-trained model)
Consolidating gists	3.3.2	Further training	Mistral-7B-Instruct-v0.2	ROC Stories dataset (Mostafazadeh et al., 2016)

Table 1: Summary of models trained. See Section 2 for further details of the training methods and model sizes, and Section 3 for further details of the tasks and training data.

3 Results

In Section 3.1 we show that the neocortical generative network learns to approximate episodic memories through consolidation, and that characteristic distortions towards the 'priors' of the network arise as a consequence, which we compare to the distortions seen in human memory (see also Section 3.3.2). In Section 3.2 we demonstrate that a consequence of the consolidation of sequential structure into the neocortical network is that it can be used for structural inference and sequential planning, and compare the characteristics of its performance to human data. In Section 3.3 we model how hippocampus and neocortex can play complementary roles in a single task, contributing memory for specifics and general knowledge respectively. In the 'full' model, we show how each event is efficiently encoded in the hippocampus as a conceptual code plus surprising details, from which the neocortex reconstructs the event during recall (Figure 1e).

3.1 Neocortical learning of narratives and consequent distortions

First we show that episodic memories can be assimilated into the neocortical network through consolidation, and that recalled narratives show distortions based on background semantic knowledge and expectations about event structure.

3.1.1 Gist-based distortions

In the Bartlett (1932) experiment, students heard a story called 'The War of the Ghosts' and were asked to recall it after different time intervals. The recalled story was distorted to be more consistent with the students' background knowledge of the world, with distortion increasing over time (Bergman & Roediger, 1999). To simulate consolidation, we fine-tuned the medium-sized GPT-2 model on the Bartlett (1932) story in addition to 'background data'. Recall of the story was explored by giving the network the first few words of the story ('One night two young men from Egulac'), and inspecting the predicted continuation.

To explore the effect of the model's 'priors' (i.e. its expectations based on previous learning) on recall of narratives, the background data distribution was varied. A dataset of Wikipedia content (Ziadé, 2024) was used, with six categories of article selected ('Politics', 'Health', 'Universe', 'Sport', 'Nature', and 'Technology'). The training data for each model was made up of 1000 articles sampled from the relevant category (with the first 1000 characters of each article taken) plus the Bartlett story. The model was trained for 5 epochs on this combined dataset to simulate consolidation, where each epoch is one full presentation of the training set. The temperature for sampling continuations from the model was also varied between 0 and 1.5 (see Section 6.3.2). The entire simulation was run five times.

When the Bartlett story is 'consolidated' into the generative network memory distortions are observed, as in the human data (see Table 2). Distortions in recalled stories reflect the 'priors' of the generative network. The word clouds in Figure 3d show that new words added to the story (i.e. 'semantic intrusions') are representative of the background dataset used, and Figure 3a and b show that the recalled stories move closer towards the background dataset in text embedding space. See also Table 3 for selected examples of semantic intrusions from the six models. Whilst distortion occurs even with 'greedy decoding', more 'semantic intrusions' are observed at higher temperatures, as Figure 3c and Table 2 show.

If the original Bartlett story is replayed enough times, the level of distortion reduces (Figure 3d). This shows that the neocortical model is able to learn the story, but contrasts with human data in which consolidation is characterised by increasing gist-based distortion (Bartlett, 1932; Payne et al., 2009). In the 'basic' model here distortion arises from assimilation into the generative network, but the distortion eventually reduces with many replays, as there is a perfect copy of the memory in hippocampus with which to train neocortex. In addition, we are testing recall as the continuation from a verbatim subsequence of the encoded memory, which can be achieved by rote memorisation alone after enough replay, but this would not be possible if recall were tested in a way that required greater 'understanding' (e.g. by asking the model questions about the story). In the 'full' model of hippocampus and neocortex (Figure 1e), the initial memory trace is not a perfect copy but is already distorted by schema-based expectations, which are further reinforced by consolidation (see Section 3.3.2). In addition, these distortions would become more apparent during consolidation as recall increasingly reflects neocortical representations rather than the hippocampal ones.

Type	Recalled story example
Original	One night two young men from Egulac went down to the river to hunt seals and while they were there it became foggy and calm. Then they heard war-cries, and they thought: “Maybe this is a war-party". They escaped to the shore, and hid behind a log. Now canoes came up, and they heard the noise of paddles, and saw one canoe coming up to them. There were five men in the canoe, and they said: . . .
Temperature of 0	One night two young men from Egulac went down to the river to hunt seals and while they were there it became foggy and calm. Then they heard war-cries, and they thought: “Maybe this is a war-party". They escaped to the shore, and hid behind a log. Now the war-party started, and they heard the noise of paddles, and saw one canoe coming up to them. There were five men in the canoe, and they said: ...
Temperature of 0.5	One night two young men from Egulac went out to the river to hunt seals and while they were there they were robbed by the men from Egulac. The men ran back to Egulac and hid behind a log. When they came out they discovered a canoe coming up to them and they jumped up and began to fight. The canoe came down to the ground and they began to wrestle. The men said they had been robbed by the thieves. . . .
Temperature of 1	One night two young men from Egulac went up to the river and they saw a canoe coming up to them. It was full speed, and they could not believe their luck when they saw it coming up the river. They jumped on the canoe and started to paddle, but the canoe was not there. And they could not feel their feet on the canoe; they thought they were sinking. When they reached the shore, they saw one canoe . . .

Table 2: Recalled stories for different temperatures, for a model trained for five epochs with the ‘Sports’ category of Wikipedia data (Ziadé, 2024) as the background data distribution, and prompted with ‘One night two young men from Egulac’.

Category	Recalled story extract
Nature	The young man went ... up the river to the summit of the glen to make a fire
	When the sun rose high above the scrub and moorland, . . .
Politics	they had been active in the local militia
	He was ... convicted for violating the law
Health	one of their companions, a young man from the neighbouring town, fell seriously ill
	“We will not go along until you are strong enough to fight"
Universe	the sun rose up in the east, and the clouds became tinged with the yellow of dawn
	A large, ferocious cataclysmic storm came out of the water and many were killed
Sport	But one of the warriors [gave] him a lesson in hand-to-hand fighting
	[They] had been training for this event for weeks
Technology	young men from Egulac went to a bar in San Francisco, California
	[He] went on to become one of the chief hunters of the seals that year.

Table 3: Extracts from recalled stories for different categories, showing how semantic intrusions reflect the ‘priors’ of the generative network. Examples are selected from a range of temperatures.

3.1.2 Event extension and contraction

Distortions can apply to the structure as well as the content of narratives. Event extension is the tendency to extend certain events in memory, while event contraction is the tendency to curtail them. Raykov et al. (2023) demonstrate this with three types of video: complete videos end at a natural event boundary, incomplete videos are curtailed before this point, and updated videos are extended beyond it. They find that incomplete videos are often extended in memory (extension errors), while updated videos are often shortened (omission errors), and show that consolidation promotes these distortions (Figure 4a.)

This simulation aims to test the hypothesis that the model captures the effect of consolidation in Raykov et al. (2023) (see Section 3.3.1 for how the immediate effects could be explained). Event extension and contraction are modelled with simple stories in text form (Rashkin et al., 2018). Three types of narrative are used. The majority are 'typical' stories, unmodified from the Rashkin et al. (2018) dataset, whereas an 'incomplete' story has characters removed, and an 'updated' story has characters added, so it continues beyond its natural ending. The 'updated' and 'incomplete' stories are all the same length. Training data consisted of 100 complete stories (to establish a 'prior' for typical stories), 10 incomplete stories, and 10 updated stories. The medium-sized GPT-2 model (Radford et al., 2019) was fine-tuned on this set of stories for five epochs and then recall was tested by giving the model the first few words and observing the output. The experiment was repeated ten times, with a different sample of stories used in each trial.

After some training, the stories are approximately memorised (recall is tested by giving the first ten words of each story as a prompt). Omission errors are more common in updated than incomplete stories, while extension errors are more common in incomplete than updated ones (Figure 4b). Omission rather than extension errors are more common overall, as in the human data. By bringing the story to a more natural conclusion or by removing an incongruous ending, the stories are made more similar to the network's expectations. See Table 4 for examples. This mirrors findings on boundary extension and contraction in memory for images (Bainbridge & Baker, 2020; Intraub & Richardson, 1989); in both cases, memories are distorted towards a prior encoded in the network by its previous 'experience', consistent with Bayesian views of memory (Hemmer & Steyvers, 2009; Spens & Burgess, 2024).

Type	**Original**	**Recalled**
Typical	Stephen wanted a cardigan for his birthday. Sally bought one for him. She wrapped it up. She wrote a note. She gave it to him! The end.	Stephen wanted a cardigan for his birthday. Sally bought one for him. She wrapped it up. She wrote a note. She gave it to him! The end.
Incomplete	Floyd was enjoying a summer day of inspecting cicada shells. He found many of them and hung them on his shirt like decorations. He squatted at a large tree's roots when he spotted a really nice one. When he picked it up, t The end.	Floyd was enjoying a summer day of inspecting cicada shells. He found many of them and hung them on his shirt like decorations. He squatted at a large tree's roots when he spotted a really nice one. He plucked it out and hung it on his shirt like a necklace. The end.
Updated	Kris lost her job. She was sad at first. She looked for a new job. She got an interview. Kris got a new job! Jack had always wanted his own dinosaur. He asked Santa Claus for one every year. He never would get one and beca The end.	Kris lost her job. She was sad at first. She looked for a new job. She got a new job! Now she makes a decent amount of money. The end.

Table 4: Event extension and contraction examples. The selected incomplete and updated stories (which are of the same length) display extension and omission errors respectively. The model is prompted with the first ten words of each story.

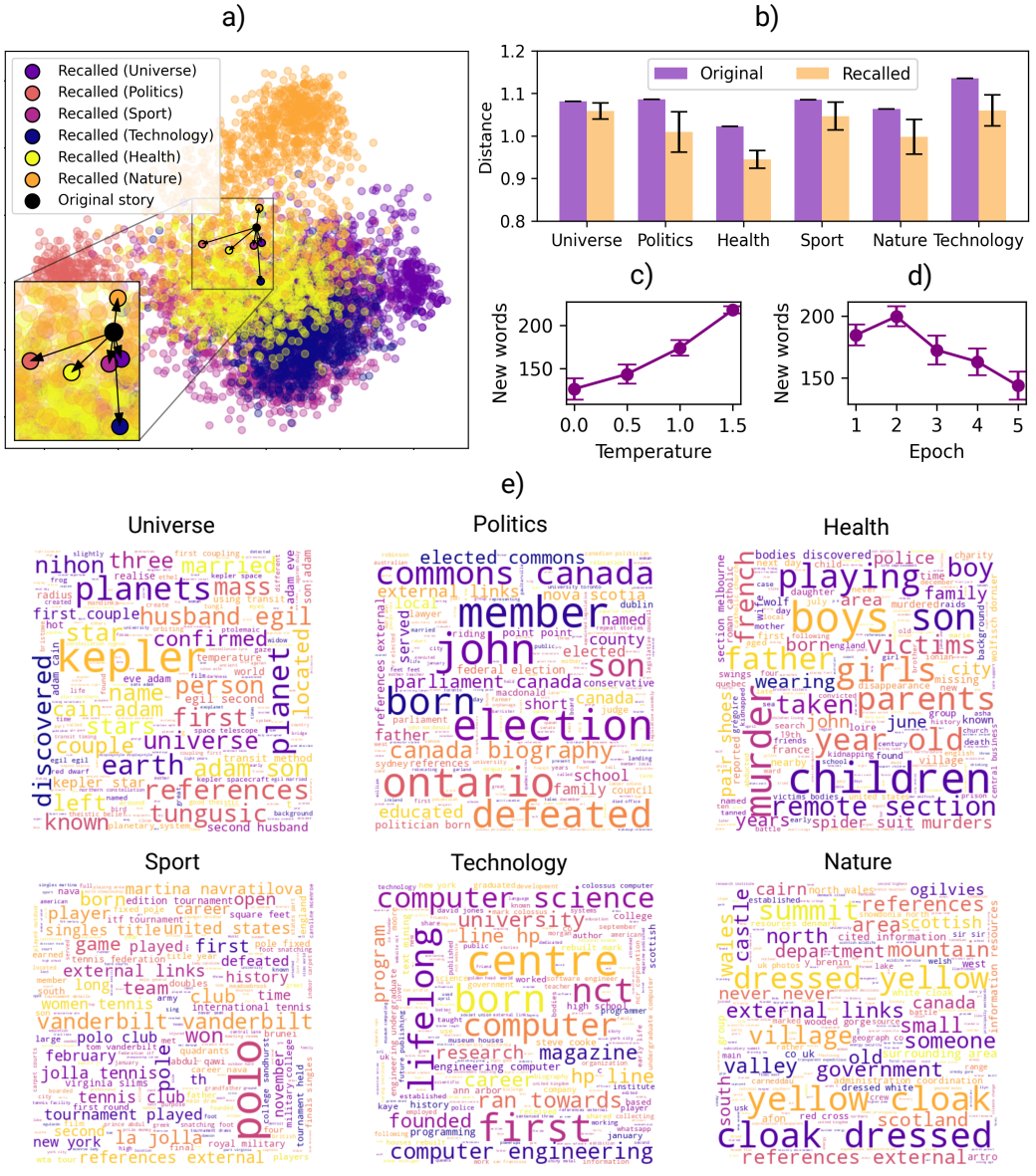

Figure 3: The effect of prior knowledge on narrative distortions. a) GPT-2 is fine-tuned on Wikipedia content (Ziadé, 2024) from each of six categories plus the Bartlett (1932) story. The embeddings of the training data plus the recalled story for each model (at a temperature of 0.5) are obtained, using 'all-MiniLM-L6-v2' from Reimers and Gurevych (2019). These embeddings are projected into 2D with PCA, showing that recall moves each story (see legend) towards its training data (points of the same colour). b) The cosine distance between the mean embedding for each category and the original story (purple) and recalled stories (orange). Recalled stories become more similar to the background dataset. Error bars give the SEM for the five recalled stories sampled (but are zero for the original stories, as there is only one). c) The effect of temperature when sampling continuations on the number of new words in the recalled Bartlett stories ('semantic intrusions'). Error bars give the SEM. All results are for the final model at epoch five. d) The effect of epoch on the number of new words in the recalled Bartlett stories, where an epoch is one presentation of the entire training dataset. This shows the effect of increased learning of the original story. Error bars give the SEM. All results use a temperature of 0.5. e) Word clouds showing 'semantic intrusions' for the six models (produced with the 'wordcloud' Python package, with common words, i.e. 'stopwords', excluded) at a temperature of 0.5, aggregated across five trials. This demonstrates that distortions reflect the model's expectations. See also Table 3.

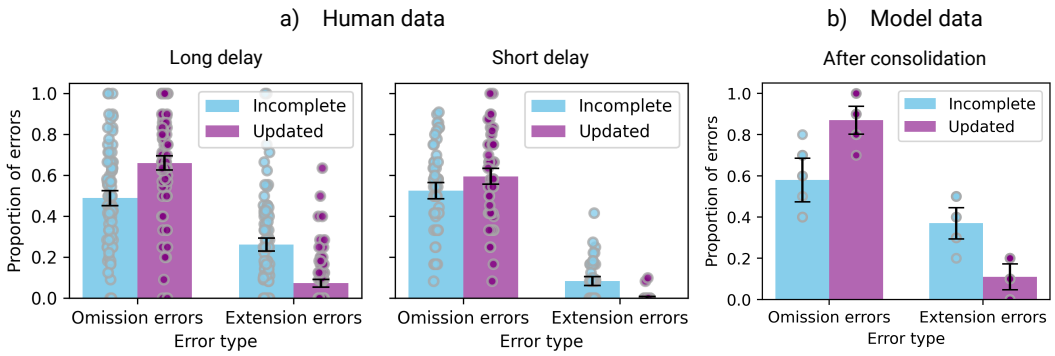

Figure 4: a) Event extension and contraction results, showing omission errors and extension errors after long (left) and short (right) retention intervals, adapted from Raykov et al. (2023) with permission. b) The mean proportion of stories recalled with omission and extension errors, averaged across ten trials, modeling performance after consolidation (cf. long delay in a). Omission errors are more common for updated than for incomplete stories, while the reverse is true for extension errors. As in part a, error bars give the 95% confidence intervals of the mean.

3.2 Neocortical learning of sequential structure for prediction and planning

Next we demonstrate how the neocortical network can extract sequential structure from memories through consolidation, developing statistical learning, structural inference, and model-based planning abilities which we compare with human data. This fits with findings that the ability to anticipate future stimuli improves with consolidation (Tarder-Stoll et al., 2024), leading to downstream improvements on a diverse range of tasks even as perceptual details fade. At a more mechanistic level, we show that the neocortical network can generalise to a new situation by aligning stimuli representations with a learned map.

3.2.1 Statistical learning

Memory consolidation is thought to support the extraction of statistical structure from experience. To explore the effect of sleep on statistical learning, Durrant et al. (2011) constructed two types of sequence of tones, one with an underlying transition structure and one without (Figure 5a). The authors found that sleep improved the recognition of structured sequences more than waking rest (Figure 5c).

To test whether the statistical learning of sequential structure through consolidation in Durrant et al. (2011) is consistent with our model, we produced a set of sequences using the experimental transition structure, representing the stimuli encoded in the hippocampus and replayed during rest. In contrast, 'unstructured' sequences were generated by randomly sampling digits between one and five. The medium GPT-2 architecture was trained from scratch for three epochs on 2000 such sequences, each made up of 50 'tones' (see Section 6.5.1, SI, for details). Perplexity is a measure of aggregated prediction error which represents the unexpectedness of a sequence given the learned statistical structure. As Figure 5d shows, the difference in perplexity between structured and unstructured sequences increases over time as the generative model is trained, in agreement with the finding that consolidation improves ability to distinguish between the two. The model becomes better at predicting the next item, enabling the generation of sequences that match the Durrant et al. (2011) transition structure (Figure 5b).

3.2.2 Structural inference

Consolidation is thought to support structural inference (Ellenbogen et al., 2007; Kumaran, 2012). A spatial example of structural inference is the finding of shortcuts, as this relies on the common structure of space, and a non-spatial example is inferring that A is the grandfather of C from the knowledge that A is the father of B, and B is the father of C, which relies on the common structure of family trees. The relations in these tasks can be seen as edges in graphs, so that multiple tasks with a

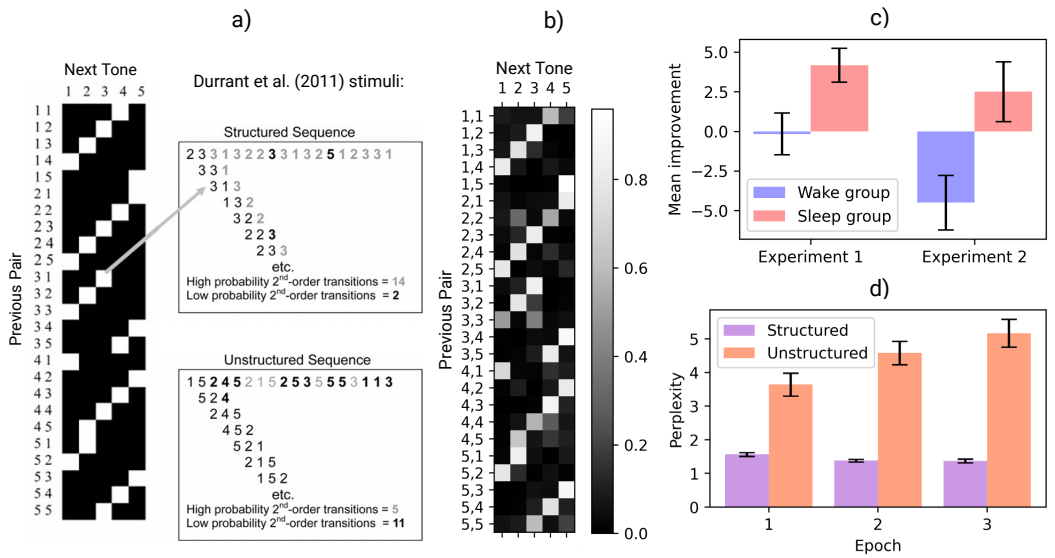

Figure 5: Statistical learning of sequential structure. a) In structured sequences of tones, the preceding two tones determine the next according to the transition matrix (left), except for 10% of transitions which are random to avoid repetition. That is, white squares indicate a 90% probability of a certain tone coming next in the sequence. In unstructured sequences, all transitions are random. Adapted from Durrant et al. (2011) with permission. b) The learned transition structure when GPT-2 is trained from scratch on data from Durrant et al. (2011), with transition probabilities extracted from data generated by the trained model (with a temperature of 0.1). c) The improvement in accuracy between immediate and delayed recall for the wake and sleep groups, adapted from Durrant et al. (2011) with permission. Two variants are tested: an overnight asleep vs. daytime awake delay (Experiment 1), and a daytime nap vs. daytime awake delay (Experiment 2). d) Perplexity (a measure of prediction error) over time for structured vs. unstructured sequences, for a model trained on structured sequences. The ability to distinguish the two types based on their perplexity increases over time. Error bars give the SEM.

common transition structure can be simulated using a fixed graph with different nodes for each task (Whittington et al., 2020).

This simulation tests the hypothesis that consolidation enables structural inference by the neocortical network. We consider inference in two types of graph: a spatial graph and a simple family tree graph (as in Whittington et al., 2020). We trained GPT-2 models on random walks on these graphs, corresponding to sequences of observations encoded in the hippocampus, and then tested the models' inference abilities on novel sequences with the same underlying structure.

In the spatial graph, a three-by-three grid represents a simple 2D environment, where the nine nodes are locations and the edges between them ('NORTH', 'EAST', 'SOUTH' and 'WEST') are possible transitions (Figure 6a). Whilst each graph's structure is the same, nodes are labelled with names (random pairs of letters) to represent arbitrary features at a particular location. Trajectories through the environment are walks on the resulting directed graph, which are represented as strings such as 'ab EAST wd SOUTH ea WEST hn'. (See graph transformers for an alternative approach; Dwivedi and Bresson, 2020.)

The family tree graph has a simple structure for illustrative purposes, consisting of two children, their parents, and two sets of grandparents. See Figure 6d. We model this as a directed graph with edges 'PARENT_OF', 'CHILD_OF', 'SPOUSE_OF', 'SIBLING_OF', 'GRANDPARENT_OF', and 'GRANDCHILD_OF'. As in the spatial case, all graphs have the same structure, but each graph has different names assigned to its nodes. Walks on the graph are represented by strings such as 'lk PARENT_OF nd SIBLING_OF re'.

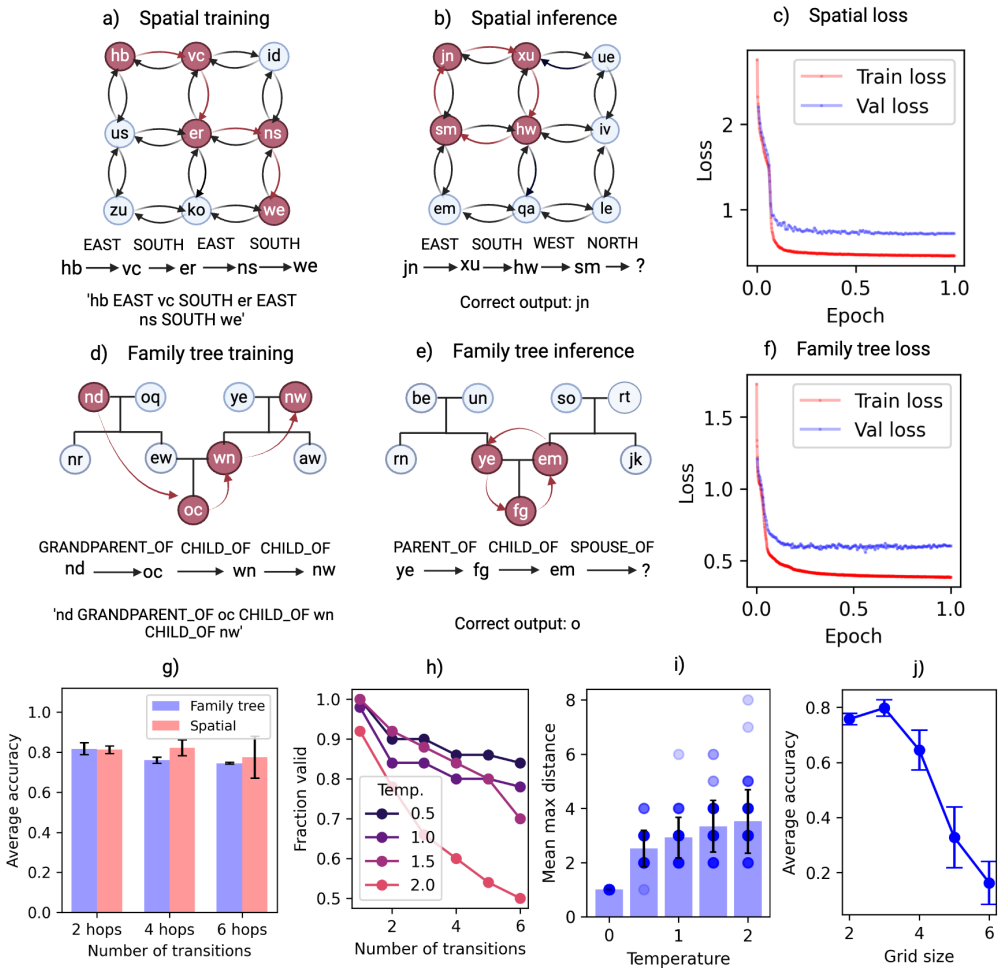

Figure 6: Learning structural regularities in graphs. a) The spatial graph task. Spatial trajectories are modelled as walks on the graph, and represented as strings as shown. b) Testing structural inference, modelled as the ability to complete sequences on unseen graphs based on structural regularities. For example, it is possible to infer that the next location in the sequence shown is 'j', given the structural regularities of spatial graphs. c) The training and validation losses of GPT-2 trained for one epoch on sequences from 500,000 spatial graphs, where the loss is a measure of aggregated prediction error. d) The family tree graph task. Relationships between family members are modelled as walks on the graph, and represented as strings as shown. e) Testing structural inference in the family tree graph. For example, it is possible to infer that the next person in the sequence shown is 'o', given the structural regularities of family tree graphs. f) The training and validation losses of GPT-2 trained for one epoch on sequences from 500,000 family tree graphs. g) Loop completion performance for the spatial and family tree models, grouped by the number of edges ('hops') in the template. Error bars give the SEM. See Tables 7 and 8, SI, for the accuracies for each template. h) The fraction of 'imagined' spatial trajectories which are structurally valid, at a range of temperatures. A sequence is marked as invalid if different locations are found at the same inferred co-ordinates, e.g. in 'sz WEST eq EAST zr' both 'sz' and 'zr' occur at the same co-ordinates, which is contradictory. i) The maximum Manhattan distance travelled from the origin for 'imagined' trajectories of length 50 tokens (with individual data points plotted, and error bars giving the SEM). j) Accuracy by grid size. The spatial inference model - trained on paths in a 3x3 grid of locations - was tested on a range of grid sizes. (The 'Grid size' axis gives the height and width of the grid in nodes, i.e. the grid size is 3 for environments in the training data.) Templates in which the final location can be inferred were populated with random locations, i.e. the test data are paths in novel environments not featuring in the training data.

In each case, we created 500,000 graphs with the same structure but randomly chosen labels (pairs of letters) for the nodes. Three random walks of 50 transitions were sampled from each graph to create the training data, representing sequences of observations that might be experienced, encoded in the hippocampus, then replayed offline. GPT-2's medium-sized architecture was then trained from scratch for one epoch (i.e. the full dataset was presented once). After training the models, we tested inference by defining a set of cycles in the graph for which the final destination should be inferred given the sequence so far. For example, the next item after 'uq NORTH sx EAST tp SOUTH ec WEST' can be inferred to be 'uq' given the structure of spatial graphs, and the next item after 'qk PARENT_OF xm PARENT_OF vw GRANDCHILD_OF zq SPOUSE_OF' can be inferred to be 'qk' given the structure of family tree graphs. (Only a subset of these tasks were tested as there are a very large number of possible loops, particularly for the family tree task.) These graph cycles were then populated with random pairs of letters, so that none of the graphs used for testing featured in the training data (i.e. the sequences used for testing were sequences of relationships in *new* family trees or spatial grids). Greedy decoding was used to generate predictions given the sequence so far. On some of the family tree inference problems there are multiple possible answers which are consistent with, but cannot be inferred from, the sequence so far (such as the imagined family member 'ef' in 'ab CHILD_OF cd PARENT_OF ef', whereas 'ab' is the expected answer), and these are counted as incorrect, making this quite a harsh performance metric.

Figure 6c and f show the 'loss' (aggregated error on the training data) of the spatial model and family tree model respectively. In both cases the loss gradually decreases, indicating improved ability to predict the next node on the set of graphs used for training, which corresponds to the consolidation of previous experience. Tables 7 and 8 and Figure 6g show good performance on a range of *novel* structural inference tasks, involving environments which have not been experienced before. Simpler inferences include inferring that going one step west then east takes one back to the original location, e.g. the correct continuation of 'ab EAST cd WEST' is 'ab'. Similarly, a correct continuation of 'ab PARENT_OF cd CHILD_OF' is 'ab'. But surprisingly complex inferences are also possible, e.g. that 'kh CHILD_OF oi SPOUSE_OF tv CHILD_OF fh SPOUSE_OF xr GRANDPARENT_OF gq SIBLING_OF' is followed by 'kh'. Furthermore given a single new item the trained spatial model can 'imagine' new trajectories consistent with the learned graph structure (Figure 6h), which explore further from the starting position at higher temperatures (Figure 6i). We also tested inference on larger grids than the model was trained on to show that inference cannot solely be attributed to 'rote memory' of sequences of transitions (Figure 6j).

To understand how the LLM generalises to a new environment, we inspected representations in its hidden layers. Activations (vectors of length 1024) for each token of a random walk were extracted from the innermost layer (i.e. the 12th of the 24 transformer blocks) of our model, and from the same layer of the pre-trained GPT-2 medium model released by Radford et al. (2019) as a baseline. Having obtained a vector for each location, we analysed the distances between these representations. We found that the nearer the points in the environment, the closer their hidden representations (Figure 7a). This is not true of the baseline model (Figure 7b). We then ran principal component analysis across representations extracted from random walks in many different environments. Figure 7d shows that our model has learned a shared structure across episodes, whereas this structure is not present in GPT-2 medium (Figure 7c). This mirrors the finding that a shared abstract representation of different spatial environments with the same structure develops in cortex through consolidation, as Baram et al. (2024) show using cross-map repetition suppression effects in fMRI. Once this abstraction of the task has developed, it supports in-context learning. The correlation between distance in the grid and distance between the representations increases with the length of a random walk on a new graph, but is high from the outset, suggesting the model can rapidly align new stimuli to its learned map (Figure 7f).

The results are consistent with the claim that consolidation of sequences into a generative model supports relational inference and generalisation. Furthermore they suggest that models trained on a simple prediction error minimisation objective can learn an abstract transition structure.

3.2.3 Model-based planning

Planning often involves mentally simulating the consequences of a sequence of actions, known as model-based planning. How does the abstraction of structure from experience through consolidation relate to this? The aim of this simulation is to test the hypothesis that generative networks trained through consolidation can support model-based planning by sequential prediction of the next item.

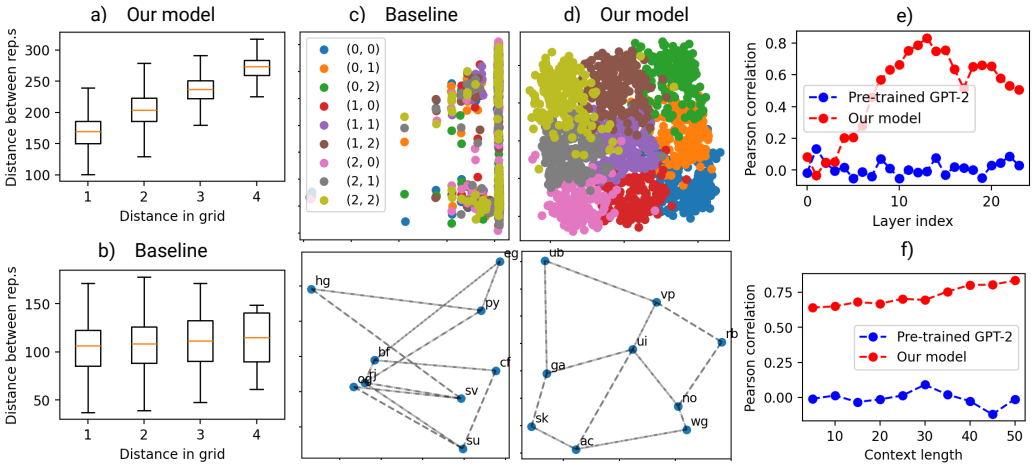

Figure 7: Exploring hidden representations in the spatial graph task. a) Activations for each token of a random walk were extracted from layer 12 of our model, in order to obtain a vector for each location (aggregating across multiple tokens where necessary). b) As above but for pre-trained GPT-2 medium as a baseline. c-d) We ran principal component analysis with two components across representations extracted from random walks in many different environments (upper row), colouring each location by its co-ordinates in the environment, so that points with the same colour are different locations which occupy the same position in the 3x3 grid (see examples in lower row). Results are shown for GPT-2 medium in c and our model in d. e) The Pearson correlation between Manhattan distance in the environment and Euclidean distance in each layer of our model (red) and GPT-2 medium (blue). f) The Pearson correlation between Manhattan distance in the environment and Euclidean distance in L12 for differing context lengths, for our model (red) and GPT-2 medium (blue).

In other words we explore whether, as another consequence of being able to (re)generate sequential experience, the network can perform 'roll-outs' enabling model-based planning.

Vikbladh et al. (2024) design a task to assess the contribution of different strategies to planning (i.e. model-based, model-free, or successor representation approaches), and explore how this changes with consolidation. As Figure 8 summarises, the task is as follows: nine items (red, green or yellow vehicles, animals or fruit) are arranged in a loop. The participants observe clockwise subsequences from the loop, determined by start, stop, and reward conditions. The start condition gives the item the sequence starts at. The stop condition gives the colour the sequence terminates at, e.g. 'green' means it stops upon reaching the first green object. The reward condition determines the object type that gives a reward of 2 (while other objects give -1). Given a particular start, stop, and reward combination, participants must decide whether to accept or reject a trial by predicting its total reward.

After learning the rules of the task, participants experience nine sequences made up of the stimuli above in a random order with one sequence per start state, each with a random stop and reward condition, making decisions then observing the sequence as feedback. Crucially, each participant sees only one stop and one reward condition for each start state during training. (Note that all participants reached a performance of at least 85% on these training sequences.) They are then tested on all combinations of stop and reward conditions without feedback, requiring them to do transition and/or reward 'revaluation' when deciding whether to accept or reject a trial with a new stop or reward condition (see Momennejad et al., 2017). Performance is tested before and after a seven day delay to assess the effect of consolidation.

The task allows model-based, model-free and successor representation strategies to be disentangled, since only model-based planning allows revaluation of both reward and transition statistics, whereas the successor representation allows only reward revaluation, and model-free planning allows neither. Vikbladh et al. (2024) found a mixture of successor representation and model-based strategies, with the model-based approach increasing with consolidation between day one and day seven (Figure 8h). MEG decoding and analysis of response times indicated that model-based planning was associated with sequential 'rollouts' involving the medial temporal lobe on both days and prefrontal cortex on day seven.

We test the hypothesis that model-based planning, based on the 'rollout' of sequences by the generative network, increases with consolidation in the neocortical network as follows. The stimuli for Vikbladh et al. (2024) can be represented as sequences of the form 'START: yellow fruit, STOP: red, REWARD: animal, SEQUENCE: green fruit (-1), red animal (2)', and it is straightforward to train the medium GPT-2 model on these sequences of text. The task is simulated as follows: i) pre-train the model on different stimuli so that it learns the rules of the task, ii) train on stimuli for a particular task with specific stop and reward conditions for each start item, representing consolidation through hippocampal replay, and iii) compare the generative model's accept / reject predictions, with new combinations of stop and reward conditions for each start item, to the correct values over the course of consolidation.

First we pre-train the model on sequences which mirror the structure but *not* the content of the task stimuli, to simulate learning the rules of the task. The pre-training data consists of the 81 possible start / stop / reward combinations for each of 1000 sets of nine items, giving a total of 81,000 sequences. Each item is a random pairing of one of three random adjectives and one of three random nouns; the start condition is one of the nine items, the reward condition is one of the three nouns, and the stop condition is one of the three adjectives. For example, one sequence might be 'START: stripy gerbil, STOP: angry, REWARD: cloud, SEQUENCE: busy cloud (2), angry plate (-1)'. The GPT-2 model is fine-tuned on this shuffled dataset for ten epochs. After this stage of training, the model can be given a prompt with randomly chosen stimuli / conditions, and generate a sequence consistent with the rules. But the model knows nothing about the Vikbladh et al. (2024) task stimuli or their order. This models the participant after rule learning, but before experiencing the task stimuli.

To simulate the task itself, the stimuli from Vikbladh et al. (2024) are used ('red animal', 'green animal', 'yellow animal', 'red vehicle', 'green vehicle', 'yellow vehicle', red fruit', 'green fruit', and 'yellow fruit'), shuffled into a random order. A single pair of stop and reward words are selected. The training stimuli are then just nine sequences with these stop and reward criteria (one per start item). These nine sequences are oversampled to 1000 items (i.e. 1000 'replayed' samples are taken, so that 9 sequences are presented 1000 times in random order) to prevent overfitting to the order of the sequences. The model is fine-tuned for eight epochs on this dataset (i.e. there are ten iterations of training on the dataset of 1000 samples), representing repeated offline replay of the training stimuli.

Three datasets are used to track performance over time: sequences requiring transition revaluation (i.e. with a new stop condition), sequences requiring reward revaluation (i.e. with a new reward condition), and sequences requiring both kinds of revaluation (i.e. with a new stop *and* reward condition). Performance on the accept / reject decision-making task is tested by obtaining the predicted sequence, and extracting the sum of the predicted rewards, accepting any trial with a sum greater than zero (see Section 6.5.3, SI). This simulation is run five times, each with a random ordering of the stimuli, reward condition, and stop condition.

Figure 8f shows that the performance on all three types of task improves over time. This suggests that the generative model learns the order of the items through 'consolidation' such that it can predict the sequence of items and rewards given novel reward and/or stop conditions. In order to explore whether planning becomes more model-based, Figure 8g compares the network's decisions to the decisions predicted by four strategies: model-free (the accept / reject decision for the same start item from the training data), reward revaluation (the decision given the ability to perform reward revaluation but not transition revaluation), transition revaluation (the decision given the ability to perform transition revaluation but not reward revaluation), and model-based (the correct decision, as deduced from a learned model of transition statistics for the task). The results in Figure 8g show that the network is consistent with the model-based planners' behaviour in Figure 8h, in that model-based planning ability increases with consolidation. The model's use of step-by-step prediction is also consistent with the model-based planners' increase in reaction time with planning depth.

3.3 Modelling hippocampal-neocortical interaction as retrieval-augmented generation

We now demonstrate how hippocampus and neocortex can play complementary roles in a single task, contributing memory for specifics and general knowledge respectively. This explains episodic contributions to inference and planning prior to full consolidation (e.g. Vikbladh et al., 2017). Furthermore, in the 'full' model memories are compressed for more efficient storage (Barlow et al., 1961; Barlow, 1989), by generating gists of events in vector form which are encoded together with unexpected details in the hippocampus (Figure 1e). This explains not only conceptual representations

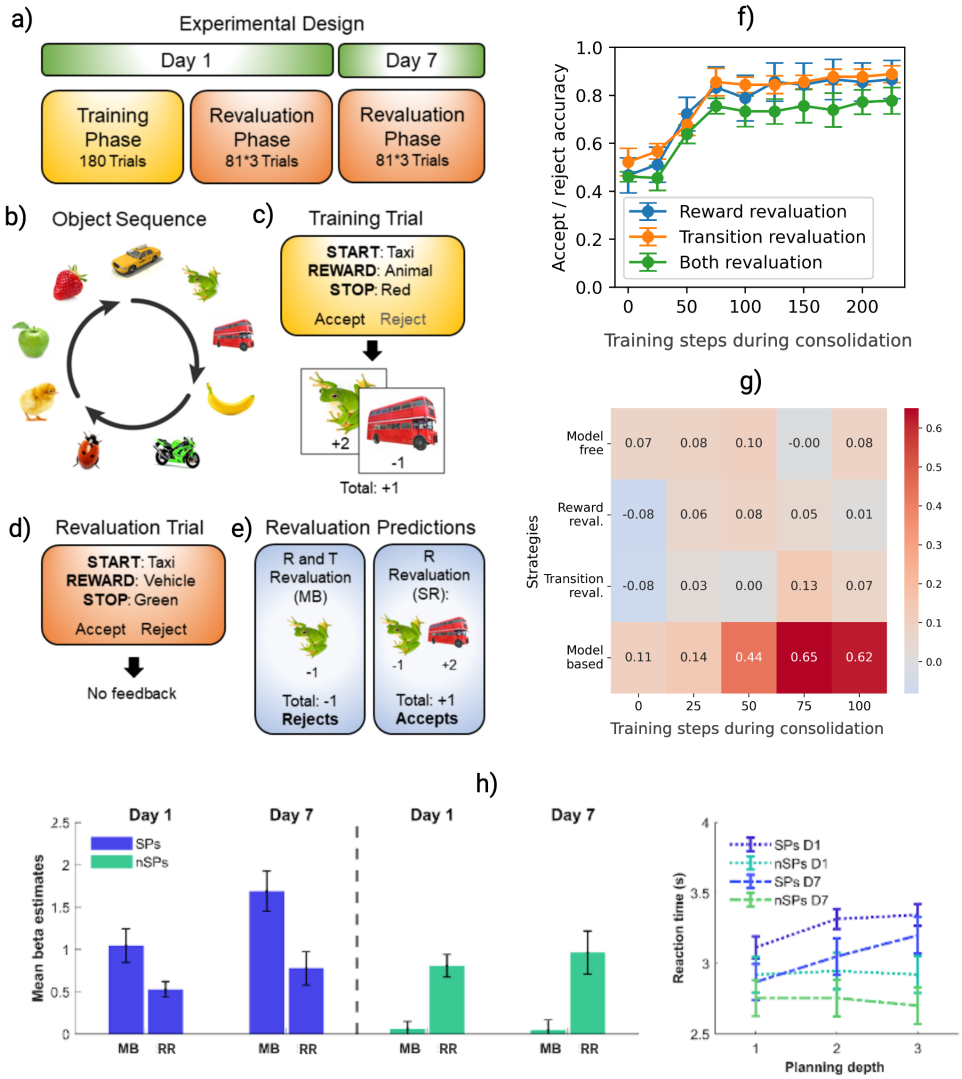

Figure 8: Parts a-e show the task design, adapted from Vikbladh et al. (2024) with permission: a) Training takes place on day one, with tests of reward and transition revaluation on days one and seven. b) The stimuli are nine items arranged in a clockwise 'loop'. c) The task is to decide whether to accept or reject a trial given a start item and reward and stop conditions, based on the expected rewards obtained from the sequence. d) The participant sees only a single reward and stop condition per start item during training, and revaluation tests performance when the reward and/or stop conditions change. e) Model-based planning (left) and the successor representation approach (right) allow different kinds of revaluation for the trial shown in d. f) Accuracy of accept vs. reject choices over the course of training on a particular trial, for three types of problem. In each case the model pre-trained on the rules of the task was further trained on just nine unique sequences for 8 epochs. The mean across five trials is taken. Errors bars give the SEM. g) Heatmaps showing regression coefficients, capturing which 'strategy' best explains the results over the course of training. The network's decisions after simulating 'consolidation' of the nine sequences are compared to the decisions if each of four strategies were used. Linear regression is used to predict the observed accept / reject decisions, with accept / reject decisions given each strategy as the regressors (i.e. multiple regression is run). h) Planning behaviour over time, adapted from Vikbladh et al. (2024) with permission. Left: Regression coefficients were obtained for model-based (MB), reward revaluation only (RR), and model-free strategies. Participants were split into two groups - sequential planners (SPs) and non-sequential planners (nSPs) - based on their beta value for the model-based strategy. SPs become more model-based with consolidation. Right: planning depth is correlated with reaction time for SPs.

in hippocampus (Quiroga, 2012), but also the fact that memories are schematised from the time of encoding (Deese, 1959; Roediger & McDermott, 1995).

3.3.1 Hippocampal and neocortical contributions to problem-solving

Memories can be used to support problem solving immediately after encoding, not just after consolidation. This simulation aims to test the hypothesis that the generative network and hippocampal network could work together to achieve this in a way resembling 'retrieval-augmented generation' (RAG; Lewis et al., 2020). Inference from recent memories is modelled as a process whereby relevant sequences from the hippocampus are retrieved and used as context for the generative model. Whilst the simulations of inference in Section 3.2.2 involved prompting the model with the sequence of observations so far, storing all past experience in working memory is not feasible given its low capacity (Baddeley, 2000; Cowan, 2001). In general, only a small subset of past episodes are relevant, so RAG provides a way to condition predictions on only the most useful information.

We create a 'toy example' of retrieval-augmented generation using the two models in the structural inference results above (one trained on spatial graphs, and one on family tree graphs, such that each has learned the structural regularities of the stimuli). In each case, the hippocampus encodes sequences from *new* spatial or family tree graphs, corresponding to observations in a new spatial environment or of a new family's relationships. Can 'missing edges' which were not present in the training data be inferred, by combining the sequences in the hippocampus with long-term knowledge of relational structure? Specifically, 100 new graphs were constructed for each task, each missing a randomly chosen edge. Short walks on each graph, which did not include the missing edge, were stored in the 'hippocampus' (simply a list of strings in this example). For each missing edge, a query of the form 'ab EAST' or 'cd PARENT_OF' was constructed for the spatial and family tree graphs respectively. In other words, if the 'cd PARENT_OF ef' edge was omitted from the graph, the test would be the model's continuation from 'cd PARENT_OF'. (Greedy decoding was used to generate predictions.)

Testing involves two stages, retrieval followed by generation: first the hippocampus is queried for relevant traces, simply by finding sequences containing the node in the query. Then the generative network produces an output conditioned on the retrieved sequence concatenated with the sequence for the task (Figure 9a), i.e. with the retrieved sequence and the current task in the LLM's prompt. The results show that this supports structural inference immediately after encoding sequences in the hippocampus, whereas relying on either the hippocampal network or generative network alone gives worse results (Figure 9b).

Planning based on memories prior to consolidation can also be modelled in this way. In Section 3.2.3, we simulated the task in Vikbladh et al. (2024) to show that the generative network learns to do model-based planning by extracting the task's transition structure through consolidation. But human participants can plan prior to consolidation too. As shown for inference, one might hypothesise that planning based on recent memories is a process of 'retrieval-augmented generation'. We assume the system involves a generative network that 'knows' the rules of the task but *not* the task stimuli, and a hippocampus which has encoded the task stimuli. As in Section 3.2.3, only 9 sequences are encoded, one per start position with a single reward and stop condition. Given a new problem (requiring a different reward condition, stop condition, or both) the system first retrieves the most relevant sequence, i.e. the one with the greatest word overlap, and then uses this to provide the context for the generative network. The generative network in Section 3.2.3 (following pre-training on the task, but before training on the actual stimuli) is used.

Figure 9c shows the accuracy of the accept vs. reject decisions with the RAG approach compared to two baselines. The 'HPC only' baseline tests the behaviour of the initial hippocampal network by itself, by copying the accept / reject prediction of the most similar stored sequence, whereas the 'NC only' baseline tests the behaviour of the generative network by itself, *prior* to any consolidation of the 9 new sequences. Both baselines are worse than the RAG approach across all three test types, suggesting the model hippocampus and neocortex play complementary roles. This could be seen as a model of episodic contributions to planning, in fitting with the finding that model-based behaviour can be supported by retrieving relevant episodic memories (Vikbladh et al., 2017).

In this simulation, only one trace is retrieved per query. But this mechanism is scalable to cases where many traces are relevant to the query, given sufficient capacity to hold them in working memory, and is thus a potential model of reasoning based on multiple recent memories as well as a single memory.

3.3.2 Combining gists and details

We propose that the hippocampus could store each memory as the combination of a conceptual representation and surprising details, with retrieval-augmented generation used to reconstruct memories from this compressed form. This minimises the hippocampal storage of redundant information that is already well-predicted by the generative network (see also Spens and Burgess, 2024). In other words, we suggest that this hippocampal representation allows the neocortical network to (re)generate experiences in a more efficient way, and that conceptual representations are optimised to enable generation.

We simulate this by drawing on recent LLM research on 'context compression', which aims to deal with the practical problems that arise from very long contexts. Approaches include summarisation of the context (Xu et al., 2023), the removal of redundant information so that only surprising elements are kept (Jiang, Wu, et al., 2023), and the transformation of the context into a vector capturing its meaning (Cheng et al., 2024; Ge et al., 2023). The xRAG approach of Cheng et al. (2024) is of particular interest, as it aligns with the evidence that hippocampal traces can include conceptual representations, and does not require any special training of the LLM itself - just additional weights that map the embedding of a sequence (i.e. the output of a 'text embedding' model) to a token in the input layer of the LLM (see Figure 2b). In brief, these weights are trained so that the embedding of the text reconstructed by the LLM is as close as possible to the embedding of the original. Another advantage is that the underlying idea in Cheng et al. (2024) is not specific to language but could in principle work for arbitrary types of sequence. Note that we refer to the xRAG vector as the 'gist vector' in the subsequent text as xRAG is one implementation of a family of approaches (see also Ge et al., 2023).

In summary, in the 'full' model the hippocampus encodes a vector representing the conceptual gist of the event in very compressed form, rather than the sequence in its entirety. Alongside this, details that are not captured by this gist vector are encoded, where the amount of detail depends on a prediction error threshold. Recalling an event involves reconstructing it from its gist vector and details with the generative network.

To illustrate how gist, detail, and semantic memory interact in the full model, we simulate the encoding and recall of narrative events as follows (Figure 10), using 100 stories from the ROC Stories dataset (Mostafazadeh et al., 2016). We use Mistral-7B-Instruct-v0.2 plus xRAG weights (released by Cheng et al., 2024) to simulate neocortex. To encode a story, its embedding is obtained from the Meng et al. (2024) text embedding model, from which the xRAG vector is derived via the xRAG weights (i.e. the xRAG weights transform the embedding into the input space of the LLM). Each story is encoded as its xRAG vector plus surprising details. To identify surprising details, all phrases are extracted from the text, then the perplexity of each phrase given the xRAG vector is calculated. In other words, for each phrase, the model is prompted with the xRAG vector and the phrase in question. Then the phrases are ranked in descending order of perplexity, with the top n phrases stored.

Given a query as the input to recall, the neocortex 'searches' the hippocampus for relevant traces. This is simulated by producing an xRAG vector for the query (as described above), and then by retrieving the trace(s) with the nearest xRAG vector. The context given to the LLM consists of the xRAG vector, the details, and a prompt describing the task. Note that the xRAG vector only occupies the space of a single token in this input. The final recalled story is the output of the LLM.

Figure 10d and Table 5 show examples of encoded and recalled stories. For instance, a story about a girl's visit to an aquarium is encoded as its gist vector and a surprising detail (that the girl took a nap on the journey). The model neocortex reconstructs an approximation of the original story from the gist vector and the detail, with a few semantic intrusions added to the story, e.g. 'Melody's favorite part was the shark tank'; these intrusions are shaped by the expectations of the model neocortex, as in Bartlett (1932). This hints at how gist, detail, and semantic memory can interact in recall.

The encoding of gist plus details was also compared to three baselines: encoding the story in full detail, encoding only the gist, and simply imagining the continuation (i.e. without using the model hippocampus at all). The mean size per memory (in tokens) and the accuracy of recall (the cosine similarity between the embeddings of the original and recalled story) were compared across the three conditions, showing a trade-off between efficiency and accuracy in memory in Figure 10e. The gist plus details encoding strikes a balance between these objectives.

In the full model, how does the level of distortion change with consolidation? An additional simulation was performed to investigate the consolidation of these representations combining gist and details, using Mistral-7B-Instruct-v0.2 (Jiang, Sablayrolles, et al., 2023). First stories from the ROC Stories dataset (Mostafazadeh et al., 2016) were encoded in gist plus details form in the model hippocampus. During consolidation, the memories were reconstructed from these traces, and the generative network was trained on the result (using low-rank adaptation; Hu et al., 2021), see SI for further details. The loss (aggregated prediction error) for the original story and for the hippocampal trace's reconstruction were calculated at the end of each epoch. The loss for the hippocampal trace decreases steadily, but the loss for the original story begins to increase after an initial drop, as the model overfits to the conceptual gist. In summary, in the full model the stored version in the hippocampus is itself an approximation, with more replay strengthening the statistical biases captured by the gist.

Original	**Gist only**	**Stored detail**	**Gist + detail**
Sarah has a bad habit. All Sarah eats is candy and sweets. She starts to notice a sharp pain in her teeth. The dentist tells her she has 6 cavities! Sarah decides she needs to start eating healthier.	Sarah has a bad habit of only eating candy and not eating healthy foods. This is a problem because she is not getting the nutrients she needs to stay healthy. Her dentist told her that she has cavities and needs to start eating more fruits and vegetables . . .	The dentist tells her she has 6 cavities.	Sarah has a bad habit of eating only candy. The dentist tells her she has 6 cavities. Sarah's bad habit of eating only candy has led to her having 6 cavities. The dentist told her this when she went for a check-up. Sarah needs to change her eating habits ...
Stan and his best friend have been friends for 8 years. They met when Stan was in first grade. They play games together, ate together, and watched movies together. Their favorite time of day is when Stan gets home from school. Stan's best friend is always waiting for him with a wagging tail.	Stan and his best friend have been friends for 8 years. They met in kindergarten and have been inseparable ever since. They do everything together, from playing in the park to watching movies at home. They even have a special handshake they use to greet each other. Stan's friend is always there for him, no matter what. . . .	Stan's best friend is always waiting for him with a wagging tail.	Stan and his best friend have been friends for 8 years. They met when Stan was walking home from school and his friend was wagging his tail and barking at him. Stan approached the dog and petted him, and they have been friends ever since. They spend a lot of time together, playing, going for walks, and just hanging out. . . .
Gina prepared food for the family. She cooked lasagna, grilled chicken and garlic bread. As soon as we started to eat, her mom became nauseous. Afterwards, everyone started to feel queasy. They eventually threw away the food and ordered Chinese food.	Gina prepared food for the family. She cooked lasagna. However, she got sick and threw up. The family ate the lasagna, but they all got sick and threw up. They went to the hospital and were diagnosed with food poisoning. The food was later found to be contaminated with *E. coli*.	Her mom became nauseous.	Gina prepared food for the family. She cooked lasagna. However, her mom became nauseous and threw up after eating it. Gina then realized that she had used spoiled ground beef for the lasagna. The rest of the family also became sick after eating it.

Table 5: Narrative examples of retrieval-augmented generation with compression as a model of hippocampal-neocortical interaction, using Mistral-7B-Instruct-v0.2 plus xRAG weights to simulate neocortex. The 'Original' column gives the story 'experienced' by the system. The 'Gist only' column gives the reconstruction from just the xRAG vector, i.e. when the xRAG vector is retrieved from the hippocampus and used to condition the generative network. The 'Stored detail' column lists the most surprising detail given the xRAG vector. Specifically, all phrases are extracted from the text, then the perplexity of each phrase given the xRAG vector is calculated. In other words, for each phrase, the model is prompted with the xRAG vector and the phrase in question. Then the phrases are ranked in descending order of perplexity, with the top phrase used as the 'Stored detail'. The text embedding (from which the xRAG vector is derived) and details for each memory are encoded together in the model hippocampus. The 'Gist + detail' column gives the reconstruction of the memory from this representation.

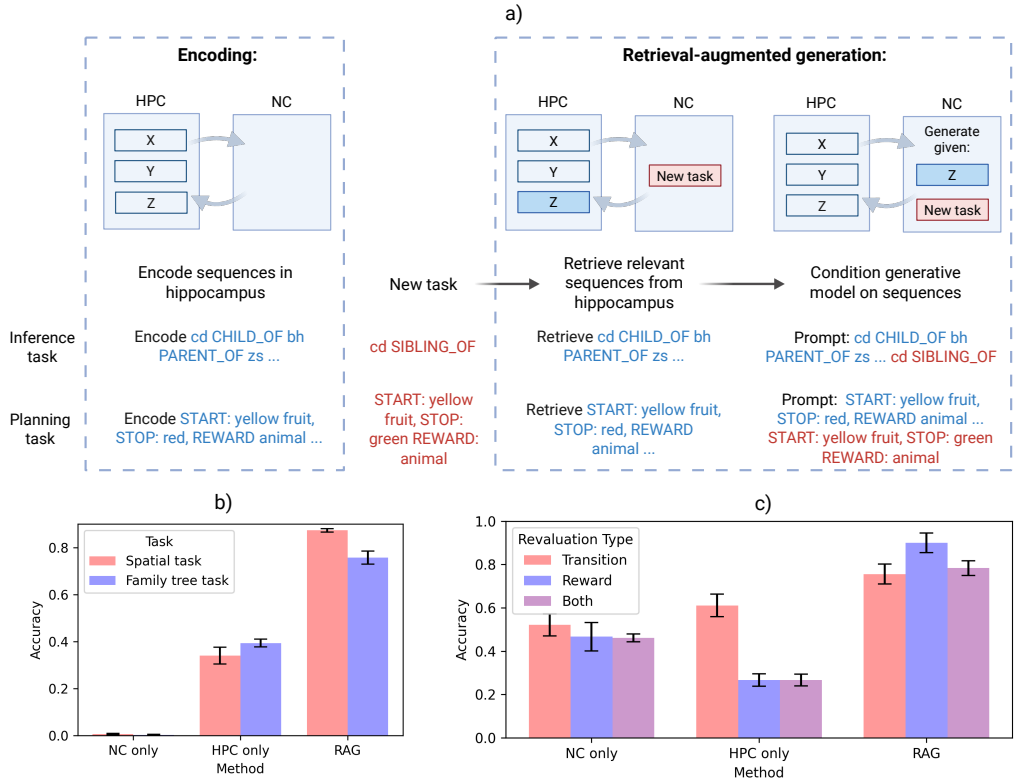

Figure 9: Retrieval-augmented generation (RAG; Lewis et al., 2020) applied to inference and planning tasks. a) The procedure for the simulations. First sequences are encoded in the hippocampus, then (before consolidation of these memories) a new task is encountered. With the RAG approach, relevant sequences are first retrieved from the hippocampus, and then provided as input to the generative network together with the new task. Specifically, in the spatial and family tree inference tasks the system encodes 100 walks from 100 different graphs (with a simplified structure, and each missing a single edge). Each encoded walk visits three nodes and two edges. The task is then for the system to infer the missing edge, e.g. predict the next location after 'ab NORTH' or 'cd SIBLING_OF' as shown, prior to any consolidation of the new graph. The models trained in Section 3.2.2 are used. To model planning *prior* to consolidation in the Vikbladh et al. (2024) task, the system encodes the nine task stimuli. When given a new task requiring revaluation, the RAG approach retrieves the most relevant sequence (the one with the same start state), and then uses this to condition the generative network. The model in Section 3.2.3 (following pre-training on the task, but before training on the actual stimuli) is used. b) Inference accuracy for the spatial and family tree inference tasks with RAG, and a 'HPC only' and 'NC only' baseline for comparison. The 'HPC only' baseline randomly selects one of the locations / people in the retrieved sequence. The 'NC only' baseline conditions the generative network on the task alone, without retrieving sequences from the hippocampus. c) Accuracy of the accept vs. reject decisions in the planning task with RAG, and a 'HPC only' and 'NC only' baseline for comparison. The 'HPC only' baseline retrieves the most similar sequence from the hippocampus and predicts the corresponding accept / reject decision. The 'NC only' baseline conditions the generative network on the task alone, without retrieving sequences from the hippocampus.

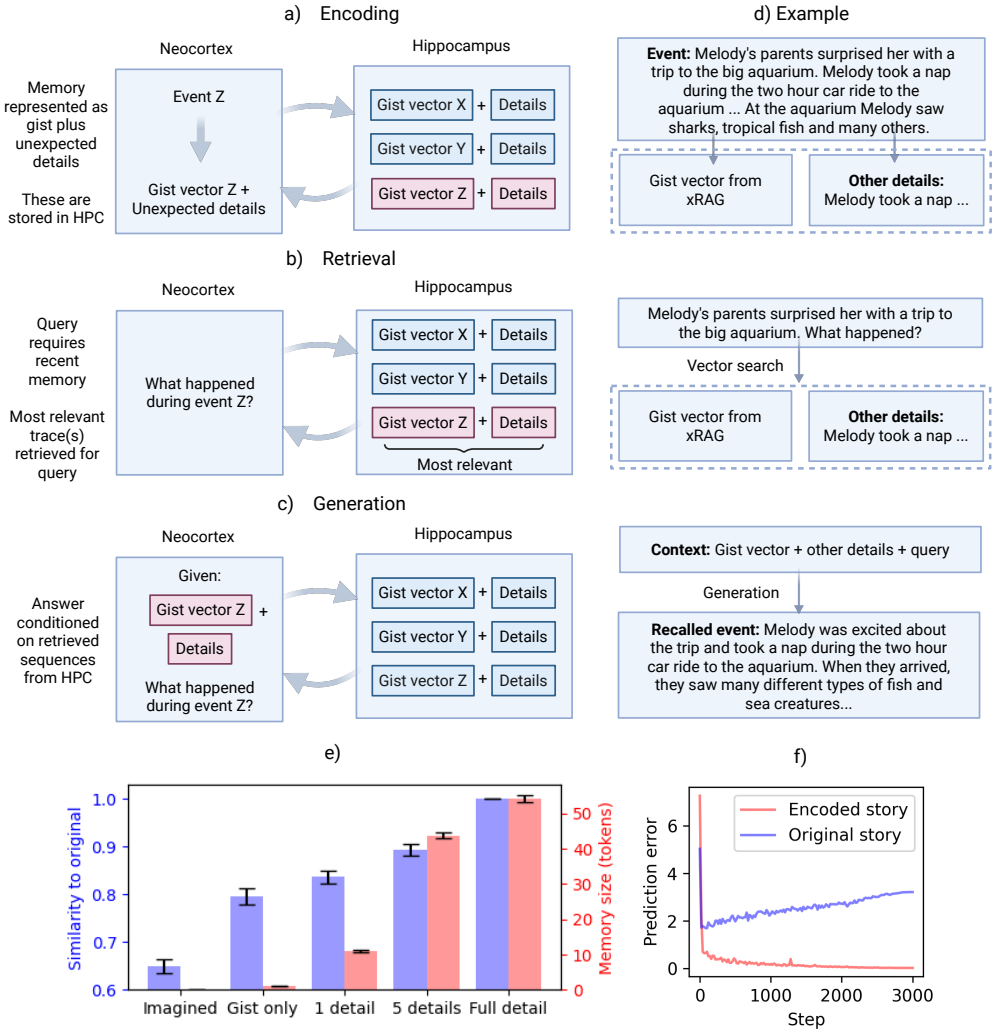

Figure 10: A model of hippocampal-neocortical interactions as retrieval-augmented generation with stories. a) The neocortex generates a gist vector for the narrative, which is encoded in the hippocampus together with unpredictable details. b) Given a query as the input to recall, the neocortex searches the hippocampus for relevant traces, e.g. by finding gist vectors in this simulation. c) The generative network in neocortex produces an 'answer' conditioned on the retrieved hippocampal trace(s). d) The stages of encoding and recall are shown for an example story from the ROC Stories dataset (Mostafazadeh et al., 2016), with Mistral-7B-Instruct-v0.2 used to simulate the neocortex, and xRAG vectors (Cheng et al., 2024) representing the conceptual gist. e) The size of the memory trace stored in the model hippocampus (in tokens) and recall accuracy (cosine similarity between original and recalled stories' embeddings with 'all-MiniLM-L6-v2'; Reimers and Gurevych, 2019) for five variants. From left to right: simply imagining the event, storing only the gist vector, storing the gist vector with one surprising detail, storing the gist vector with five surprising details, and storing the event in full. f) When the gist plus details representation is consolidated into the model neocortex (here, Mistral-7B-Instruct-v0.2), the loss of the encoded story steadily decreases, but the loss of the original story increases after an initial drop. Here the loss is a measure of aggregated prediction error (which can be thought of as 'surprise').

4 Discussion

We have presented a model of the construction and consolidation of sequential memories, in which sequences encoded in the hippocampus are replayed to train a generative network to capture the statistics of experience through prediction error minimisation. This model suggests an explanation for several features of human cognition, such as the bidirectional influences between episodic and semantic memory; semantic memory is acquired by learning to reconstruct episodic memories, but also shapes the representation of these episodes, both in terms of their initial encoding, as reflected by the compressed representations observed in hippocampus (Quiroga, 2012), and their gradual schematisation (Bartlett, 1932). To support this, we demonstrated how distortions arise in recalled narratives as neocortex learns to approximately reconstruct them, and how these reflect priors in the generative model derived from previous experience (Sections 3.1 and 3.1.2). Our model also explains how a number of capabilities improve through consolidation, in addition to the memorisation of 'replayed' sequences. In particular, the neocortical generative network supports statistical learning of transition probabilities (Section 3.2.1), inferring new relationships from limited observations by learning an implicit structure (Section 3.2.2), and model-based sequential planning (Section 3.2.3).

The model suggests a mechanism for how task-relevant episodic memories are integrated with general knowledge. We propose that the neocortical generative network interacts with stored sequences retrieved from the hippocampus as retrieval-augmented generation (Lewis et al., 2020), but with the 'generator' gradually trained on memories from the 'retriever' through consolidation, and the 'retriever' storing memories in a highly compressed form that can be decoded by the 'generator'. This is consistent with our ability to use memories for complex tasks such as planning and inference, which cannot be supported by the retrieval of traces alone, even prior to consolidation (e.g. Vikbladh et al., 2024). Furthermore the stored traces in the hippocampal 'retriever' can combine surprising details with gists in vector form, binding together information at different levels of abstraction in memory, with the conceptual representation optimised for reconstruction (Section 3.3.2). This strikes a balance between efficient storage and accurate recall, and explains how neocortical expectations shape memories from their encoding onwards (e.g. Bartlett, 1932; Raykov et al., 2023).

In recent years, neuroscience has seen a move from a modular view of many semi-independent networks learning particular tasks to a focus on the learning of multipurpose representations. Similarly, there has been a transition in machine learning from task-specific models to larger task-general ones trained through self-supervised learning, sometimes referred to as 'foundation models' (Bommasani et al., 2021). The use of a self-supervised task to train such models is key to their scaling, as external supervision is not required. We can think of the brain as learning neural 'foundation models' too, and this paper suggests how memory consolidation could contribute to their development (see also Spens and Burgess, 2024). This view aligns with the growing consensus that prediction error minimisation is key to biological intelligence (e.g. Friston, 2010), neuroimaging evidence of large task-general networks (e.g. Raichle et al., 2001), and the relative lack of external supervision in learning.

We note that other types of model like recurrent neural networks (RNNs) could perhaps perform some of the simpler tasks tested in this paper. Using LLMs, despite their greater complexity, has the following advantages. Firstly, linguistic stimuli are commonly used to test episodic and semantic memory, and transformer-based models enable the simulation of these tasks in a way simpler models cannot. Secondly, the concept of 'retrieval-augmented generation' (Lewis et al., 2020) in the LLM literature sheds light on how neocortical and hippocampal networks could work together to solve complex tasks based on a combination of general and specific information. Thirdly, LLMs have rich internal representations that could be compared to neural data (see Figure 7). In short, while RNNs can also generate sequences, the sophistication of LLMs allows a wider range of phenomena to be considered, even though their implementation in the brain would require a more biologically plausible learning rule (see Section 4.1). We do not argue that modern LLMs mirror the brain, but that some kind of neural 'foundation model' being trained to minimise prediction error on replayed sequences provides a good model for a wide range of data on memory, consolidation, inference and sequential planning.

The following sections discuss limitations of the model and directions for future research.

4.1 Future work

This work is primarily a model of psychological, rather than neural, data. More could be done to bridge the gap between these ideas and the growing understanding of sequence representations at a neural level. For example, the model allows consideration of navigation, which is associated with distinctive cell types such as place, grid and head-direction cells, so the connection between these different levels of explanation could be explored (see Banino et al., 2018; Schönfeld and Wiskott, 2015; Sorscher et al., 2023; Stachenfeld et al., 2017; Whittington et al., 2020). Furthermore, progress in 'mechanistic interpretability' (e.g. Bricken et al., 2023) enables the hidden layer activations of LLMs to be compared to neuroimaging data, allowing comparison of representations at a population level as well.

Modern language models like GPT-2 are trained on a simple next item prediction task (Radford et al., 2019), mirroring how prediction error minimisation is thought to drive biological learning (e.g. Friston, 2010). But even if the objective is shared, the implementations may be very different. In particular, error backpropagation - which calculates each parameter's 'responsibility' for the network's error - is thought to be implausible because of its non-local nature (Whittington & Bogacz, 2019). More biologically plausible approximations to error backpropagation do exist (Whittington & Bogacz, 2019), as well as other families of network that mimic the brain more closely (Dayan et al., 1995; Friston, 2010; Rao & Ballard, 1999), which could be explored in future work.

Our model captures the sequential nature of experience, but in reality each moment is 'high-dimensional', more like the images in Spens and Burgess (2024) or Fayyaz et al. (2022) than the 'tokens' used here. Video data, whilst still far simpler than reality, better reflect this: a video is made up of a succession of frames, each of which is rich in data, and recalling it requires 'filling in the gaps' of a particular frame as well as predicting successive ones. This could be modelled by reproducing the current item (as in in Fayyaz et al., 2022; Spens and Burgess, 2024) as well as predicting the next, which would require changes to the associative and generative networks. Since pattern completion of both the current stimulus and the next stimulus would occur in the associative network, a combination of autoassociative and heteroassociative connectivity might be required (Sompolinsky & Kanter, 1986). In addition, the generative model could be replaced with one for video data (e.g. Yan et al., 2021 or Ha and Schmidhuber, 2018), although the time and cost required to train such a model might make this impractical.

Previous work suggests that the generative network may involve an encoder in neocortex, inferring a latent representation from sensory inputs, and a decoder returning to sensory cortex via the hippocampal formation (Spens & Burgess, 2024). This is consistent with the continued dependence on the hippocampal formation for many kinds of episode formation (e.g. McKenzie and Eichenbaum, 2011; Nadel and Moscovitch, 1997), and accounts for semantic memory as direct projections from the latent variables. In this paper we do not address the neural substrates of different components of the generative network, but GPT-2 does not have an encoder-decoder architecture, and aligns less closely to our view of the anatomy. In other words, models like GPT-2 do not have explicit latent variables (unlike, for example, variational autoencoders; Kingma and Welling, 2013), but learn a conceptual representation more implicitly. Current failure modes of LLMs have been linked to the lack of an explicit world model (Bender et al., 2021), so one might speculate that future variants will capture this more concretely.

The full model presented here addresses the mixed composition of memories but only scratches the surface of how this mixture changes over time. Firstly, as the hippocampal trace decays, or becomes redundant through consolidation to neocortex, the contribution of unique details relative to conceptual gist reduces. Secondly, the already distorted hippocampal trace is consolidated, reinforcing the schematic predictions used to encode it. Thirdly, consolidation continues until a trace is approximately, not perfectly, remembered, at which point the schema-congruent content may be learned to a greater degree than the schema-incongruent details (see Tse et al., 2007). Fourthly, re-encoding the gist of a memory in light of new information introduces further biases (e.g. Loftus and Palmer, 1974). More extensive simulations could explore the interaction of these factors.

Whilst experience is continuous, it is discretised into events in memory (Newtson, 1973; Newtson & Engquist, 1976), a process known as event segmentation. Event segmentation is thought to be based on prediction error (Franklin et al., 2020; Kumar et al., 2023; Zacks et al., 2007). Whilst the sequences used in our simulations are 'pre-segmented' for simplicity, this is potentially consistent with our model, as the generative network can provide an ongoing measure of 'surprise' during

perception (Franklin et al., 2020; Kumar et al., 2023; Michelmann et al., 2025), e.g. as quantified by perplexity (Radford et al., 2019). One complexity is that event segmentation occurs at multiple levels of granularity (Zacks et al., 2007); this could potentially involve averaging prediction error over different time periods, or alternatively variable error thresholds (so that more fine-grained segmentation occurs when the error exceeds a lower threshold, and more coarse-grained segmentation a higher threshold). However how a hierarchy of segments of varying lengths would be stored is unclear.

Human learning can display leaps of insight (e.g. Kounios and Beeman, 2014), with abrupt improvements thought to correspond to the reorganisation of mental representations (Ohlsson, 1992). Future work could explore the development of an LLM's internal representations as a task is learned, comparing this to the corresponding 'behavioural' changes, to see if sudden increases in performance correspond to sudden representational changes. This could then be compared to the dynamics of learning in humans.

Future research could also explore how retrieval, encoding, and replay processes are orchestrated. We simplify matters by using pre-segmented sequences, encoding every event with a fixed prediction error threshold, and 'hard-coding' when the hippocampus is queried. However, one could train a 'meta-controller' to co-ordinate these processes with reinforcement learning. This might learn to use different cognitive resources when it was optimal to do so, e.g. prioritising memories for replay to maximise future performance, and only encoding memories when it is worth the cognitive cost. This could build on previous studies like Lu et al. (2022), which prototypes these ideas with simpler models.

Finally, many aspects of human cognition are inextricably linked to language. Some of the computational modelling approaches in this paper could be applied more broadly to studies involving narratives (e.g. Bransford et al., 1972; Loftus and Palmer, 1974). A number of mental health symptoms and conditions are thought to relate to maladaptive processing of narratives, such as rumination (Nejad et al., 2013), delusion (Coltheart et al., 2011), and confabulation (Kopelman, 2010), so this may be fruitful from a computational psychiatry perspective.

4.2 Episodic and semantic memory

Generative networks such as large language models (LLMs) can memorise 'event-unique' specifics as well as generalities in a single network (Carlini, Ippolito, et al., 2022), raising questions for the common assumption that episodic and semantic memory must depend on different computational processes. If a series of narratives representing 'episodes' are 'consolidated' into an LLM, the resulting LLM could support both memory for specific episodes and for semantic 'facts', with the latter learned as a side-effect of reconstructing the former. Not only are 'beliefs' influenced by 'episodes' the network was trained on, but the 'episodes' are reshaped by the 'beliefs', as we saw in a range of memory distortion simulations. This can be the case even for the initial hippocampal encoding of episodes in the full model, as demonstrated in Section 3.3.2.

How do we distinguish remote memory from imagination, if both consist largely of schema-based predictions from a generative model? The generative model's prediction error for sequences from the training data tends to be lower than for novel sequences (Carlini, Chien, et al., 2022), so the generative network could provide a measure of recognition memory even if the original hippocampal trace has faded away. However this might correspond to a 'sense of familiarity' rather than definitive knowledge that a stimulus was experienced. Thus, the presence of similar real memories (Deese, 1959; Roediger & McDermott, 1995), or the rehearsal of imagined events (Loftus & Pickrell, 1995), can induce false memory, presumably from a 'sense of familiarity' in the generative network. This could align with hippocampal amnesics' deficits in recognition memory but preserved ability to make relative judgements of familiarity (Migo et al., 2009).

On the other hand, episodic memories arguably need to include some details from a hippocampal trace to be recognised confidently (see Yonelinas et al., 2005). Memories may retain some limited trace in the hippocampus proper for a long time, so the retrieval of event-unique details which have not yet been consolidated could help to distinguish real memories from imagination. The neocortical network may contain many semantic facts relating to one's personal history, e.g. that one went on a holiday to a particular place as a child, and could thus construct events consistent with these facts. However, if the feeling of episodic recollection requires retrieval of unique details encoded during

the event, then even an accurate reconstruction would not qualify as a episodic memory, with the implication that a memory could lose its 'episodic' status without its reported content changing.

4.3 Conclusion

We have proposed a model in which hippocampal replay trains a neocortical generative network to support the (re)construction of sequences in memory, extending previous work on static images (Fayyaz et al., 2022; Spens & Burgess, 2024) to sequential experiences, and shedding light on how memory enables prediction of the future (Schacter et al., 2007). We show that, by simply learning to predict the next item, the neocortical network can support statistical learning, inference, and model-based sequential planning, while capturing the schema-based distortions in memory for narratives. Inspired by 'retrieval-augmented generation' (Lewis et al., 2020), we show how the hippocampal and neocortical networks could jointly contribute to problem solving, allowing memories retrieved from the hippocampus to provide the context for prediction using the 'general knowledge' of the neocortical network. Furthermore, to efficiently encode sequential experiences, the hippocampal network combines unique details with conceptual representations which capture the gist of events. These representations are optimised for neocortical reconstruction, allowing the generative network to fill in the outline of a memory with its predictions.

References

Alammar, J. (2018). The illustrated transformer [Accessed on March 19, 2024]. https://jalammar.github.io/illustrated-transformer/

Alammar, J. (2019). The illustrated GPT-2 [Accessed on March 19, 2024]. https://jalammar.github.io/illustrated-gpt2/

Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, *2*.

Baddeley, A. (1992). Working memory. *Science*, *255*(5044), 556–559.

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

Baddeley, A. (2001). The concept of episodic memory. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *356*(1413), 1345–1350.

Bainbridge, W. A., & Baker, C. I. (2020). Boundaries extend and contract in scene memory depending on image properties. *Current Biology*, *30*(3), 537–543.

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M. J., Degris, T., Modayil, J., et al. (2018). Vector-based navigation using grid-like representations in artificial agents. *Nature*, *557*(7705), 429–433.

Baram, A., Nili, H., Barreiros, I., Samborska, V., Behrens, T. E., & Garvert, M. M. (2024). An abstract relational map emerges in the human medial prefrontal cortex with consolidation. *bioRxiv*, 2024–10.

Barlow, H. B., et al. (1961). Possible principles underlying the transformation of sensory messages. *Sensory communication*, *1*(01), 217–233.

Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, *1*(3), 295–311.

Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology*. Cambridge university press.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.

Bergman, E. T., & Roediger, H. L. (1999). Can Bartlett's repeated reproduction experiments be replicated? *Memory & cognition*, *27*(6), 937–947.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bransford, J. D., Barclay, J. R., & Franks, J. J. (1972). Sentence memory: A constructive versus interpretive approach. *Cognitive psychology*, *3*(2), 193–209.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., … Olah, C. (2023). Towards monosemanticity: Decomposing language models with dictionary learning [https://transformer-circuits.pub/2023/monosemantic-features/index.html]. *Transformer Circuits Thread*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Burgess, N., & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological review*, *106*(3), 551.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., & Tramer, F. (2022). Membership inference attacks from first principles. *2022 IEEE Symposium on Security and Privacy (SP)*, 1897–1914.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., & Zhang, C. (2022). Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Chaudhry, H. T., Zavatone-Veth, J. A., Krotov, D., & Pehlevan, C. (2023). Long sequence Hopfield memory. *arXiv preprint arXiv:2306.04532*.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, *34*, 15084–15097.

Cheng, X., Wang, X., Zhang, X., Ge, T., Chen, S.-Q., Wei, F., Zhang, H., & Zhao, D. (2024). Xrag: Extreme context compression for retrieval-augmented generation with one token. *arXiv preprint arXiv:2405.13792*.

Coltheart, M., Langdon, R., & McKay, R. (2011). Delusional belief. *Annual review of psychology*, *62*, 271–298.
Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, *24*(1), 87–114.
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.
Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz machine. *Neural computation*, *7*(5), 889–904.
Deese, J. (1959). On the prediction of occurrence of particular verbal intrusions in immediate recall. *Journal of experimental psychology*, *58*(1), 17.
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
Doll, B. B., Duncan, K. D., Simon, D. A., Shohamy, D., & Daw, N. D. (2015). Model-based choices involve prospective neural activity. *Nature neuroscience*, *18*(5), 767–772.
Durrant, S. J., Taylor, C., Cairney, S., & Lewis, P. A. (2011). Sleep-dependent consolidation of statistical learning. *Neuropsychologia*, *49*(5), 1322–1331.
Dwivedi, V. P., & Bresson, X. (2020). A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.
Ellenbogen, J. M., Hu, P. T., Payne, J. D., Titone, D., & Walker, M. P. (2007). Human relational memory requires time and sleep. *Proceedings of the National Academy of Sciences*, *104*(18), 7723–7728.
Fayyaz, Z., Altamimi, A., Zoellner, C., Klein, N., Wolf, O. T., Cheng, S., & Wiskott, L. (2022). A model of semantic completion in generative episodic memory. *Neural Computation*, *34*(9), 1841–1870.
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.
Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature reviews neuroscience*, *11*(2), 127–138.
Ge, T., Hu, J., Wang, L., Wang, X., Chen, S.-Q., & Wei, F. (2023). In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.
Ha, D., & Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.
Hemmer, P., & Steyvers, M. (2009). A Bayesian account of reconstructive memory. *Topics in Cognitive Science*, *1*(1), 189–202.
Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of mathematical psychology*, *46*(3), 269–299.
Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
Intraub, H., & Richardson, M. (1989). Wide-angle memories of close-up scenes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*(2), 179.
Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023). Mistral 7b. https://arxiv.org/abs/2310.06825
Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y., & Qiu, L. (2023). Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.
Káli, S., & Dayan, P. (2000). Hippocampally-dependent consolidation in a hierarchical model of neocortex. *Advances in Neural Information Processing Systems*, *13*.
Káli, S., & Dayan, P. (2002). Replay, repair and consolidation. *Advances in Neural Information Processing Systems*, *15*.
Kambhampati, S., Valmeekam, K., Guan, L., Stechly, K., Verma, M., Bhambri, S., Saldyt, L., & Murthy, A. (2024). Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*.
Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.
Kopelman, M. D. (2010). Varieties of confabulation and delusion. *Cognitive neuropsychiatry*, *15*(1-3), 14–37.
Kounios, J., & Beeman, M. (2014). The cognitive neuroscience of insight. *Annual review of psychology*, *65*(1), 71–93.

Krotov, D., & Hopfield, J. (2020). Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*.

Krotov, D., & Hopfield, J. J. (2016). Dense associative memory for pattern recognition. *Advances in neural information processing systems*, *29*.

Kumar, M., Goldstein, A., Michelmann, S., Zacks, J. M., Hasson, U., & Norman, K. A. (2023). Bayesian surprise predicts human event segmentation in story listening. *Cognitive science*, *47*(10), e13343.

Kumaran, D. (2012). What representations and computations underpin the contribution of the hippocampus to generalization and inference? *Frontiers in Human Neuroscience*, *6*, 157.

Lengyel, M., & Dayan, P. (2007). Hippocampal contributions to control: The third way. *Advances in neural information processing systems*, *20*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Li, D., Rawat, A. S., Zaheer, M., Wang, X., Lukasik, M., Veit, A., Yu, F., & Kumar, S. (2022). Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*.

Loftus, E. F., & Palmer, J. C. (1974). Reconstruction of automobile destruction: An example of the interaction between language and memory. *Journal of verbal learning and verbal behavior*, *13*(5), 585–589.

Loftus, E. F., & Pickrell, J. E. (1995). The formation of false memories.

Lu, Q., Hasson, U., & Norman, K. A. (2022). A neural network model of when to retrieve and encode episodic memories. *elife*, *11*, e74445.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., & Bossan, B. (2022). Peft: State-of-the-art parameter-efficient fine-tuning methods.

Marr, D. (1970). A theory for cerebral neocortex. *Proceedings of the Royal society of London. Series B. Biological sciences*, *176*(1043), 161–234.

Marr, D. (1971). Simple memory: A theory for archicortex. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *262*(841), 23–81.

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.

McKenzie, S., & Eichenbaum, H. (2011). Consolidation and reconsolidation: Two lives of memories? *Neuron*, *71*(2), 224–233.

Meng, R., Liu, Y., Joty, S. R., Xiong, C., Zhou, Y., & Yavuz, S. (2024). Sfr-embedding-mistral:enhance text retrieval with transfer learning. https://www.salesforce.com/blog/sfr-embedding/

Michelmann, S., Kumar, M., Norman, K. A., & Toneva, M. (2025). Large language models can segment narrative events similarly to humans. *Behavior Research Methods*, *57*(1), 1–13.

Migo, E., Montaldi, D., Norman, K. A., Quamme, J., & Mayes, A. (2009). The contribution of familiarity to recognition memory is a function of test format when using similar foils. *Quarterly Journal of Experimental Psychology*, *62*(6), 1198–1215.

Millidge, B., Salvatori, T., Song, Y., Lukasiewicz, T., & Bogacz, R. (2022). Universal Hopfield networks: A general framework for single-shot associative memory models. *International Conference on Machine Learning*, 15561–15583.

Millidge, B., Seth, A., & Buckley, C. L. (2021). Predictive coding: A theoretical and experimental review. *arXiv preprint arXiv:2107.12979*.

Momennejad, I., Russek, E. M., Cheong, J. H., Botvinick, M. M., Daw, N. D., & Gershman, S. J. (2017). The successor representation in human reinforcement learning. *Nature human behaviour*, *1*(9), 680–692.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., & Allen, J. (2016). A corpus and cloze evaluation for deeper understanding of commonsense stories. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 839–849.

Nadel, L., & Moscovitch, M. (1997). Memory consolidation, retrograde amnesia and the hippocampal complex. *Current opinion in neurobiology*, *7*(2), 217–227.

Nagy, D. G., Orban, G., & Wu, C. M. (2025). Interplay of episodic and semantic memory arises from adaptive compression.

Nagy, D. G., Török, B., & Orbán, G. (2020). Optimal forgetting: Semantic compression of episodic memories. *PLoS Computational Biology*, *16*(10), e1008367.

Nejad, A. B., Fossati, P., & Lemogne, C. (2013). Self-referential processing, rumination, and cortical midline structures in major depression. *Frontiers in human neuroscience*, *7*, 666.

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of personality and social psychology*, *28*(1), 28.

Newtson, D., & Engquist, G. (1976). The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology*, *12*(5), 436–450.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. *Advances in the psychology of thinking*, *1*, 1–44.

Payne, J. D., Schacter, D. L., Propper, R. E., Huang, L.-W., Wamsley, E. J., Tucker, M. A., Walker, M. P., & Stickgold, R. (2009). The role of sleep in false memory formation. *Neurobiology of learning and memory*, *92*(3), 327–334.

Quiroga, R. Q. (2012). Concept cells: The building blocks of declarative memory functions. *Nature Reviews Neuroscience*, *13*(8), 587–597.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., & Shulman, G. L. (2001). A default mode of brain function. *Proceedings of the national academy of sciences*, *98*(2), 676–682.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, *2*(1), 79–87.

Rashkin, H., Bosselut, A., Sap, M., Knight, K., & Choi, Y. (2018). Modeling naive psychology of characters in simple commonsense stories. *arXiv preprint arXiv:1805.06533*.

Raykov, P. P., Varga, D., & Bird, C. M. (2023). False memories for ending of events. *Journal of Experimental Psychology: General*.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084

Robin, J., & Moscovitch, M. (2017). Details, gist and schema: Hippocampal–neocortical interactions underlying recent and remote episodic and spatial memory. *Current opinion in behavioral sciences*, *17*, 114–123.

Roediger, H. L., & McDermott, K. B. (1995). Creating false memories: Remembering words not presented in lists. *Journal of experimental psychology: Learning, Memory, and Cognition*, *21*(4), 803.

Salvatori, T., Song, Y., Hong, Y., Sha, L., Frieder, S., Xu, Z., Bogacz, R., & Lukasiewicz, T. (2021). Associative memories via predictive coding. *Advances in Neural Information Processing Systems*, *34*, 3874–3886.

Samborska, V., Butler, J. L., Walton, M. E., Behrens, T. E., & Akam, T. (2022). Complementary task representations in hippocampus and prefrontal cortex for generalizing the structure of problems. *Nature Neuroscience*, *25*(10), 1314–1326.

Schacter, D. L., Addis, D. R., & Buckner, R. L. (2007). Remembering the past to imagine the future: The prospective brain. *Nature reviews neuroscience*, *8*(9), 657–661.

Schönfeld, F., & Wiskott, L. (2015). Modeling place field activity with hierarchical slow feature analysis. *Frontiers in computational neuroscience*, *9*, 51.

Sompolinsky, H., & Kanter, I. (1986). Temporal association in asymmetric neural networks. *Physical review letters*, *57*(22), 2861.

Sorscher, B., Mel, G. C., Ocko, S. A., Giocomo, L. M., & Ganguli, S. (2023). A unified theory for the computational and mechanistic origins of grid cells. *Neuron*, *111*(1), 121–137.

Spens, E., & Burgess, N. (2024). A generative model of memory construction and consolidation. *Nature Human Behaviour*, 1–18.

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

Sun, W., Advani, M., Spruston, N., Saxe, A., & Fitzgerald, J. E. (2023). Organizing memories for generalization in complementary learning systems. *Nature neuroscience*, *26*(8), 1438–1448.

Tang, M., Barron, H., & Bogacz, R. (2023). Sequential memory with temporal predictive coding. *arXiv preprint arXiv:2305.11982*.

Tarder-Stoll, H., Baldassano, C., & Aly, M. (2024). Consolidation enhances sequential multi-step anticipation but diminishes access to perceptual features. *Psychological Science*, 09567976241256617.

Tse, D., Langston, R. F., Kakeyama, M., Bethus, I., Spooner, P. A., Wood, E. R., Witter, M. P., & Morris, R. G. (2007). Schemas and memory consolidation. *Science*, *316*(5821), 76–82.

Vikbladh, O., Burgess, N., & Russek, E. (2024). Systems consolidation of sequential dynamics in model-based planning. *bioRxiv*.

Vikbladh, O., Shohamy, D., & Daw, N. (2017). Episodic contributions to model-based reinforcement learning. *Annual conference on cognitive computational neuroscience, CCN*.

Warstadt, A., & Bowman, S. R. (2022). What artificial neural networks can tell us about human language acquisition. In *Algebraic structures in natural language* (pp. 17–60). CRC Press.

Whittington, J. C., & Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in cognitive sciences*, *23*(3), 235–250.

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

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xu, F., Shi, W., & Choi, E. (2023). Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Yan, W., Zhang, Y., Abbeel, P., & Srinivas, A. (2021). VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*.

Yonelinas, A. P., Otten, L. J., Shaw, K. N., & Rugg, M. D. (2005). Separating the brain regions involved in recollection and familiarity in recognition memory. *Journal of Neuroscience*, *25*(11), 3002–3008.

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.

Ziadé, T. (2024). Wikipedia topics [Accessed: 2024-05-31].

5 Acknowledgements

We thank Oliver Vikbladh for his advice on the planning simulations, and Misun Kim, Steve Fleming, Kris Jensen, and Mathias Sablé-Meyer for their insightful comments on the manuscript.

Funding support for this work was received from a Wellcome Principal Research Fellowship 'Neural mechanisms of memory and prediction: Finding structure in experience' (222457/Z/21/Z) (N.B.) and a Wellcome Collaborative Award 'Organising knowledge for flexible behaviour in the prefrontal-hippocampal circuitry' (214314/Z/18/Z) (N.B.).

6 Supplementary information

6.1 Data availability

The following datasets (all covered by the Creative Commons Attribution 4.0 License) were used in the simulations:

Dataset	**Origin**
Wikipedia Topics (Ziadé, 2024)	https://huggingface.co/datasets/wikipedia-topics
ROC Stories (Mostafazadeh et al., 2016)	https://cs.rochester.edu/nlp/rocstories

Table 6: Overview of data availability.

6.2 Code availability

Code for all simulations can be found at https://github.com/ellie-as/sequence-memory.

Some diagrams were created using BioRender.com.

6.3 Further model details

6.3.1 Modelling sequence learning

The simulations described above use autoregressive sequence models to represent the generative networks trained through hippocampal replay. Specifically GPT-2 (Radford et al., 2019), a transformer-based deep neural network for text generation, is used.

The primary goal during training is to adjust the model's parameters through maximum likelihood estimation, so that the probability it predicts for the true next item in each sequence, based on the items so far, is as high as possible. In other words, the network's weights are updated to predict the probability distribution of the next item as accurately as possible, a process known as causal language modelling. The training data for the original GPT-2 model is the WebText dataset of online content. Once the model is trained, it predicts the probability distribution across all items given the items so far. To generate a sequence one can sample from this distribution or simply take the most probable item at each step. The equation below gives the probability of a sequence x as a product of conditional probabilities of its items:

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, ..., s_{n-1})$$

We use the Transformers Python library (Wolf et al., 2019) in all simulations. The first stage of using GPT-2 (and similar models) is to prepare the inputs with tokenisation. A tokeniser maps commonly occurring chunks of characters to IDs (in order to look up the right token embedding in a learned embedding matrix); in the case of language tokens are often words or parts of words. The concept of tokenisation is applicable to arbitrary sequences, but for simplicity and consistency across the simulations all stimuli are converted to strings of characters (if they are not already text-based). Except for the simulation in Section 3.2.1, in which a custom tokeniser is fitted to the dataset, the same tokeniser is used as in the pre-trained GPT-2 model.

What exactly does causal language modelling with a custom dataset involve? First the training data is split into blocks, and then for every block the cross-entropy loss is aggregated across all the next token prediction tasks within the block. For GPT-2, the block size (which is also the context size of the trained model) is 1024 tokens. This means that the model is trained to consider up to 1024 tokens of context when predicting the next token in a sequence. So for each block the model tries to predict the second token based on the first token, then the third token based on the first two, and so on, until it predicts the final token based on the preceding 1023.

The loss measures the difference between two probability distributions: the distribution predicted by the model and the actual distribution in the data. For each token prediction task, the actual distribution is a 'one-hot' vector with a one for the real next token and zeros elsewhere. Specifically, the cross-entropy loss for a single prediction task is calculated as the negative log probability assigned by the model to the actual next token. For a block of tokens, the total loss is the sum of the cross-entropy losses for each token prediction task within the block, and the weights of the model are updated based on this total loss. This procedure is the same whether the model is fine-tuned or trained from scratch.

The learning rate decreases over the course of training in all simulations, as per the default behaviour of the Trainer class in the Transformers Python library (Wolf et al., 2019).

6.3.2 Sampling options

There are many ways to generate sequences from a trained sequence model like GPT-2. The simplest is greedy decoding, where the model selects the token with the highest probability as the next token in the sequence. However this can lead to repetitive and predictable sequences. Sampling from the learned probability distribution with a given temperature introduces randomness into the selection of the next item, and provides a way to control the model's 'imaginativeness'. Specifically, the temperature parameter determines the 'sharpness' of the distribution from which output tokens are selected, so that sequences at a higher temperature are more 'imaginative', but more likely to be nonsensical.

The equation below describes the 'softmax with temperature' function that is applied to the vector of scores for each token to produce a vector of *probabilities*. A large temperature T flattens the distribution, whereas T close to zero approximates a 'one-hot' vector, with a probability of one for the most likely token:

$$\sigma(s_i) = \frac{e^{\frac{s_i}{T}}}{\sum_{j=1}^{n} e^{\frac{s_j}{T}}}$$

Unless stated otherwise, sequences are sampled from the probability distribution of tokens with a given temperature. Beam search is a search strategy used in some simulations, which considers multiple potential paths through the model's probability 'landscape'. At each step, it keeps a fixed number (the beam width) of the most probable sequences generated so far and extends them, eventually choosing the sequence with the highest overall probability.

6.3.3 Further details on attention

In short, the attention mechanism works as follows: a query, a key, and a value vector are produced for each element in the sequence by learned weights. For a given element, the model computes a set of attention scores which determine how much attention should be paid to each element of the input sequence when representing this element. This is achieved by taking the dot products of the given element's query vector with the key vectors for each element. The final output is the sum of the value vectors for each element weighted by the corresponding attention scores (with some normalisation), allowing the model to aggregate information from the most relevant parts of the input. This describes a single 'attention head', but GPT-2's architecture features multiple attention blocks, each with multiple attention heads, which tend to develop different 'specialisms' through training. To be more precise the equation is given below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

In this equation, Q, K and V represent the query, key, and value matrices, respectively. These are derived from the input data. d_k represents the dimensionality of the keys (and queries), and is used to scale the dot products in a way that leads to more stable gradients. The softmax function is applied to ensure the weights sum up to 1. When the resulting term is matrix multiplied by V, the result is the relevance-weighted sum of the vector representations in V, for each token in V. In other words, the attention mechanism enriches a token's representations with a relevance-weighted sum of the other tokens' representations. See Alammar (2018) for a detailed description of this process.

Self-attention is a special case of attention in which elements of a sequence 'attend to' the sequence they are a part of. Specifically, 'masked' self-attention is used, meaning that the representation of a token in an attention block only 'attends to' preceding tokens. Previous models like BERT (Devlin et al., 2018) perform much less well at text generation partly because their attention is not masked in this way.

Transformer blocks come in different varieties. GPT-2 (Radford et al., 2019) is a decoder-only model consisting of a stack of transformer decoder blocks. Each such block consists of a masked self-attention layer followed by a feedforward layer. In a nutshell, inference with GPT-2 works as follows: First an embedding for each token is obtained from a learned embedding matrix. A key point is that position embeddings representing the tokens' positions in the sequence are added to the token embeddings (these are required because self-attention is permutation-invariant, so otherwise information about the order of items in the sequence would be lost). Then the sequence of embeddings passes through a series of transformer decoder blocks, each further enriching the token representations so that they come to capture the meaning of the text. See Alammar (2019) for helpful illustrations of the stages of processing in GPT-2. GPT-2 (Radford et al., 2019) comes in several sizes, with the number of transformer blocks depending on the size of the model. The small variant has 12 transformer blocks, the medium variant has 24 transformer blocks, and the large variant has 36 transformer blocks.

6.4 Supplementary results

6.4.1 Modelling inference

Tables 7 and 8 give the inference accuracy for each template tested in Section 3.2.2.

Inference template	**Mean accuracy**
{} EAST {} WEST {}	0.87
{} WEST {} EAST {}	0.82
{} NORTH {} SOUTH {}	0.81
{} SOUTH {} NORTH {}	0.87
{} EAST {} SOUTH {} WEST {} NORTH {}	0.71
{} SOUTH {} WEST {} NORTH {} EAST {}	0.78
{} WEST {} NORTH {} EAST {} SOUTH {}	0.8
{} NORTH {} EAST {} SOUTH {} WEST {}	0.81
{} EAST {} EAST {} NORTH {} WEST {} WEST {} SOUTH {}	0.79
{} NORTH {} NORTH {} WEST {} SOUTH {} SOUTH {} EAST {}	0.74

Table 7: Spatial task inference templates and their average accuracies.

6.4.2 Compressing sequences

When considering alternative options for how the brain might store sequences efficiently, we could draw inspiration from compression algorithms applied to sequences. For example, linear predictive coding in signal transmission involves fitting a linear predictive model to a set of sequences, and then transmitting the parameters of the model, followed by the deviations from its predictions. This enables the signal to be accurately reconstructed while transmitting far less information (see Millidge

Inference template	Mean accuracy
{} CHILD_OF {} PARENT_OF {}	0.81
{} PARENT_OF {} CHILD_OF {}	0.76
{} GRANDCHILD_OF {} GRANDPARENT_OF {}	0.8
{} GRANDPARENT_OF {} GRANDCHILD_OF {}	0.62
{} CHILD_OF {} CHILD_OF {} GRANDPARENT_OF {} SIBLING_OF {}	0.7
{} CHILD_OF {} SPOUSE_OF {} PARENT_OF {} SIBLING_OF {}	0.75
{} PARENT_OF {} SIBLING_OF {} CHILD_OF {} SPOUSE_OF {}	0.79
{} PARENT_OF {} PARENT_OF {} GRANDCHILD_OF {} SPOUSE_OF {}	0.77
{} CHILD_OF {} SPOUSE_OF {} CHILD_OF {} SPOUSE_OF {} GRANDPARENT_OF {} SIBLING_OF {}	0.75
{} GRANDPARENT_OF {} SIBLING_OF {} CHILD_OF {} SPOUSE_OF {} CHILD_OF {} SPOUSE_OF {}	0.72

Table 8: Family tree task inference templates and their average accuracies.

et al., 2021 for a review). Table 9 shows examples of simulating the compression of sequences in a similar way; only a subset of items *not* predicted by the generative network need to be stored (together with their position in the sequence) to reconstruct the sequence.

Model	Event	Trace
Pre-trained GPT-2	What lovely weather! I went for a walk in the park and the sun was shining.	What lovely weather! I went for _ _ _ _ park _ the _ _.
Model from Section 3.2.3, pre-trained on random stimuli with the structure in Vikbladh et al. (2024)	START: yellow fruit, STOP: green, REWARD: animal, SEQUENCE: red animal (2), green vehicle (-1)	_ _ _ yellow fruit _ _ _ green _ _ _ _ animal _ _ _ _ red animal _ _ green vehicle _ _ _
Model from Section 3.2.2, trained on random walks on spatial graphs	ab EAST bn SOUTH ty NORTH bn NORTH iu	ab E _ bn SOU _ ty NOR _ _ _ NOR _ iu

Table 9: Compressing sequences with generative models. The 'Model' column gives the model used to compress the sequence, the 'Event' column gives the original sequence, and the 'Trace' column gives its compressed version, which is sufficient to guarantee accurate reconstruction of the 'event' if the generative network does not change. Crucially, the tokens replaced by underscores do not need to be stored, as they are correctly predicted by the relevant model. For example, in the second row 'START:', 'STOP:', 'REWARD:', and 'SEQUENCE:' do not need to be stored as the model has learned that they occur in every sequence in the training data. Furthermore, the rewards do not need to be stored as the model has learned to predict these from the sequence so far, making them redundant.

6.4.3 Modelling sequence memory in the hippocampus

The mechanisms by which sequences are stored as traces in the initial hippocampal network are not the focus of this paper, and the hippocampus is not simulated explicitly in the results above. However, we now outline how the hippocampus might store sequences in the sequential model, in a way that supports replay and recall prior to consolidation.

In the static model (Spens & Burgess, 2024), a modern continuous Hopfield network (MCHN; Ramsauer et al., 2020) represents the hippocampus, interpreted such that the feature units activated by an event are bound together by a memory unit (Krotov & Hopfield, 2020). In the sequential model, this could be adapted to store sequences as follows. Let us consider sequences represented as strings

of characters or symbols, which can capture language, spatial trajectories, transitions in a graph, stimuli in sequential learning tasks, and more.

Millidge et al. (2022) give a unifying account of how neural network models of associative memory such as MCHNs operate, which helps to explain the extension to sequences. They develop the biologically plausible modern Hopfield network (Krotov & Hopfield, 2016) into 'a general framework for understanding the operation of ... memory networks as a sequence of three operations: similarity, separation, and projection', which they term 'universal Hopfield networks' (Millidge et al., 2022, Abstract). They observe that any of these models can be made asymmetric or heteroassociative rather than autoassociative, if the projections from feature units to memory units capture the current state, but the projections from memory units back to feature units are different (for sequences, these projection weights would correspond to the 'next state'). Then one state retrieves the next, rather than each state retrieving a pattern-completed version of itself. That is, we want to store patterns X and 'next patterns' X_{next} instead of just patterns, and change the MCHN's update rule accordingly:

$$\xi_{next} = X_{next} softmax(\beta X^T \xi)$$

The modern continuous asymmetric Hopfield network (MCAHN) converts the MCHN to work for sequences, consistent with the framework above (Chaudhry et al., 2023). However there is a problem when you try to apply this naively to certain sequences, as the MCAHN is by default a 'Markov chain' model of sequential memory. Whilst it works for sequences in which items are unique, it doesn't work for, e.g., sentences without modification, as it cannot accurately remember sequences with repeated items; see Figure 5a of Tang et al. (2023). When an item is repeated in a sequence, Tang et al. (2023) show that a MCAHN has trouble recalling the next item, instead producing a composite.

Tang et al. (2023) introduce the temporal predictive coding network (tPC), a more complex network learnt by gradient descent, based on the intuition that higher layers minimize prediction error at lower layers (Salvatori et al., 2021). (A drawback of the tPC is that the model must be 'presented with the sequence for multiple epochs' until convergence, so it is no longer capable of one-shot learning, which arguably undermines its plausibility as a model of hippocampal encoding.) They show that the one-layer tPC also has the MCAHN's issue with repeated items, but suggest the two-layer tPC to resolve this. As Tang et al. (2023) show in Figure 5d of their paper, the hidden layer represents an item's position in the sequence so the next item can be recalled correctly.

Alternatively, one could combine the concept of the modern asymmetric Hopfield network (with return projections from memory to feature units representing the next state) with a state representation that captures the history. This is based on previous work involving the role of temporal context in memory (Burgess & Hitch, 1999; Howard & Kahana, 2002). Consider the example of encoding sentences, i.e. sequences of characters. Suppose each state is a vector of length equal to the number of symbols, consisting of one at the index of the current symbol, plus the previous state multiplied by some decay factor. See Figure 11 for an example. This has the benefit of still being compatible with one-shot learning, although the memory capacity may not scale as well as the more complex predictive coding model approach. Initial testing suggests that this model can encode and retrieve sentences (although retrieval performance goes down with sentence length).

To show how the preceding 'context' is captured, consider the example of encoding sentences, i.e. sequences of characters. Each state can be represented by a vector of length equal to the number of symbols, with one at the index of the current symbol, plus the previous state multiplied by some decay factor. In other words, we want a vector representation $\mathbf{x_i}$ of the ith state which is a sum of the current symbol's vector $\mathbf{v}$ and activity from the previous state $\mathbf{x}_{i-1}$ multiplied by a decay rate λ (noting that the states are then normalised to unit length):

$$\mathbf{x}_i = \frac{\mathbf{v}_i + \lambda \cdot \mathbf{x}_{i-1}}{\|\mathbf{x}_i\|}$$

As with all modern Hopfield network (Ramsauer et al., 2020) variants, the inverse temperature, β, is a key parameter determining the network's behaviour. In this case a high value of β would be desirable to avoid composite states. The softmax function is given below; in the limit of high β, the output of this function becomes a vector with one at the index of the maximum and value and zero elsewhere:

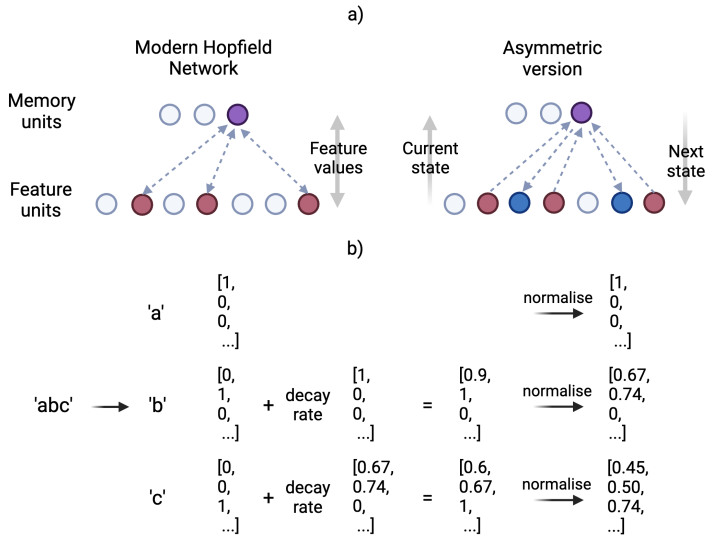

Figure 11: Using asymmetric modern Hopfield networks to model the hippocampus. a) In asymmetric versions of modern Hopfield networks, the weights between feature and memory units differ, as visualised from the perspective of Krotov and Hopfield (2020). b) A sequence of arbitrary symbols, e.g. 'abc', is represented as a sequence of vectors over the feature units, where each vector is the sum of the current state plus the decay rate (here set to 0.9) times the previous state. For each state, the return connections from memory units to feature units encode the next state.

$$softmax(z_i) = \frac{e^{\beta z_i}}{\sum_{j=1}^{K} e^{\beta z_j}}$$

Whilst this simple model helps justify the assumption that the hippocampus can store sequences associatively, it clearly has some limitations. For example, it suggests that when a state from a certain point in an encoded sequence triggers recall, states preceding that point cannot be recalled (only the subsequent states). This is clearly not the case, so a model of sequence memory that can only retrieve states in one temporal direction is unsatisfactory. Also, if states involve vectors of continuous rather than binary variables, the decaying activity from previous states could be hard to distinguish from the current state, so a more complex solution could be required. In addition, the sequences consist of a single symbol at any moment in time, but other sequences like frames in a video consist of complex representations at each time step. The Hopfield network variant could perhaps have a combination of autoassociative and heteroassociative connectivity to account for this, with autoassociative connections reconstructing the current time step, and heteroassociative connections activating the next (see Sompolinsky and Kanter, 1986).

6.5 Supplementary methods

Table 10 summarises the hyperparameters and data volumes used in each simulation in which GPT-2 was trained.

6.5.1 Statistical learning

In this simulation, sequences of tones were represented as comma-separated lists of numbers, with each number corresponding to a single tone.

At the end of each epoch of the training, the perplexity was calculated for test sets of 100 structured and 100 unstructured sequences. Perplexity is a measure of aggregated prediction error which represents the unexpectedness or schema incongruency of a sequence given the learned statistical structure; the greater the perplexity of the unstructured compared to structured test sequences, the greater the ability to distinguish them.

Simulation	Training method	Epochs	Learning rate	Batch size	Training data	Trials
Gist-based distortions	Further training of pre-trained GPT-2 medium	5	5e-04	1	Bartlett story plus 1000 Wikipedia extracts of 1000 characters each	5
Event extension and contraction	Further training of pre-trained GPT-2 medium	5	5e-04	1	100 typical, 10 incomplete, and 10 updated stories from ROCStories	10
Statistical learning	Training GPT-2 medium architecture from scratch	3	1e-05	1	2000 sequences, each made up of 50 'tones'	3
Structural inference (spatial)	Training GPT-2 medium architecture from scratch	1	5e-05	1	Three random walks of length 50 on each of 500,000 graphs	1
Structural inference (family trees)	Training GPT-2 medium architecture from scratch	1	5e-05	1	Three random walks of length 50 on each of 500,000 graphs	1
Planning (rule model)	Further training of pre-trained GPT-2 medium	10	5e-05	1	81,000 sequences (corresponding to 1000 sets of stimuli)	1
Planning (particular trials)	Further training of 'rule model' (see above)	8	5e-05	1	Nine unique sequences oversampled to 1000 sequences	5

Table 10: Summary of training hyperparameters and data volumes for each simulation in which GPT-2 was trained.

6.5.2 Event extension and contraction

Event extension and contraction are modelled with simple stories in text form, each only a few lines long (Rashkin et al., 2018). Three types of narrative are used. The majority are 'typical' stories, unmodified from the Rashkin et al. (2018) dataset, whereas an 'incomplete' story has characters removed, and an 'updated' story has characters added, so it continues beyond its natural ending. Note that this is a challenging task as some stories could be interpreted as either incomplete or updated. To disentangle the effect of story category from the effect of length, the incomplete and updated stories were all the same length. To achieve this, text from a different random story was added to stories shorter than average, extending them to the mean length, while longer than average stories were shortened to the mean length.

6.5.3 Model-based planning

Performance on the accept / reject decision-making task is tested as follows:

1. Obtain the predicted sequence from the model given an input of the form 'START: yellow animal, STOP: red, REWARD: animal, SEQUENCE:' via greedy decoding.
2. Extract the predicted rewards (any numbers in round brackets) from the sequence.

3. If the sum of the predicted rewards is greater than zero, the decision is to 'accept' the trial, and if the sum is less than zero, the decision is to 'reject' it. (If the sum equals zero, the two decisions are equivalent, so the outcome variable is set to 0.5, midway between 0 for 'reject' and 1 for 'accept'.)
4. Calculate the accuracy of the accept / reject decisions by comparing the predictions with the true values. The 'true' accept vs. reject decisions are obtained from the extracted rewards as described above.

6.5.4 Consolidation of conceptual representations

The model in Figure 10f is fine-tuned in a different way because of its size.

Instead of fine-tuning the entire model, parameter-efficient fine-tuning (PEFT) methods update only a small subset of parameters. One such method is low-rank adaptation (LoRA; Hu et al., 2021), which injects trainable low-rank matrices alongside frozen pre-trained weights. We implement LoRA with the 'PEFT' Python library (Mangrulkar et al., 2022), applying LoRA to the attention blocks alone.