Learning sequence models through consolidation

Eleanor Spens [1] **Neil Burgess** [1]

Abstract

Episodic memory is a reconstructive process, thought to depend on schema-based predictions made by generative models learned through systems consolidation. We extend previous work on memory for static scenes to model the construction and consolidation of sequential experience. After sequences are encoded in the hippocampus, a network is trained to predict the next item in a sequence during replay (simulated by training GPT-2 on a range of stimuli). The resulting model can memorise narratives, with characteristic gist-based distortions, and can also be applied to non-linguistic tasks such as spatial and relational inference. In addition, we explore 'retrieval augmented generation', in which sequence generation is conditioned on relevant 'memories', as a model for how hippocampal specifics can be combined with neocortical general knowledge.

1. Introduction

Brains are thought to make predictions in order to survive. Previous research explores how memories may be replayed by the hippocampus over the course of systems consolidation to train a generative or predictive model of the world, which supports multiple cognitive functions including episodic memory, semantic memory, imagination, and inference (Spens & Burgess, 2024; Fayyaz et al., 2022; Káli & Dayan, 2000; 2002). This provides a more mechanistic account of the theory that episodic memories are reconstructions influenced by our beliefs, i.e. that recall involves 'predicting' the past (Hemmer & Steyvers, 2009).

Recent progress on large language models (LLMs) demonstrates that complex behaviours can develop as a byproduct of a simple 'next item prediction' task (Radford et al., 2019; Brown et al., 2020). Crucially, LLMs can memorise specific sequences as well as learning generalities (Carlini et al., 2022), meaning that episodic and semantic information can be captured in a single network. In this paper we model consolidation as the training of autoregressive sequence models (simulated using GPT-2; Radford et al., 2019) on hippocampal memories. We show that our model displays similar gist-based memory distortions to those observed in human data (Bartlett, 1932), and is capable of classic spatial / relational inference tasks (Whittington et al., 2020), consistent with findings that consolidation promotes inference and generalisation (Ellenbogen et al., 2007; Kumaran, 2012).

Many situations require inference, planning, and generalisation based on recent memories, *before* their content has been assimilated into the generative network. This seems to require both specific knowledge from episodic memories in the hippocampus *and* more general knowledge from the neocortical world model (Robin & Moscovitch, 2017). Extensive dialogue between hippocampus and neocortex is observed when recalling memories (Norman et al., 2021), suggesting that these networks can contribute jointly to some tasks. Retrieval augmented generation (Lewis et al., 2020) typically refers to an approach for combining an LLM with a dataset of other information; given a query, relevant data is retrieved from the dataset, and used to prompt the LLM. One might hypothesise that neocortical generative models and more veridical hippocampal representations could be combined in a similar way, with neocortical generations conditioned on hippocampal traces.

2. Methods

GPT-2 (Radford et al., 2019) is a deep neural network which can be trained on arbitrary linguistic or non-linguistic sequences. Training involves learning to predict the next item. Once the model is trained, it can continue from an input sequence, or generate a new sequence, by iteratively predicting the probability distribution for the next item.

The following simulations use GPT-2 (Radford et al., 2019) to represent the generative networks trained through hippocampal replay. Specifically, the medium-sized model with 345 million parameters is used. In Section 3.2, the GPT-2 architecture is trained from scratch with randomly initialised weights, while in Section 3.1 existing GPT-2 weights are used as the starting point for further training. The hippocampus is not modelled explicitly, but the training data for the generative networks is intended to represent

[1]University College London. Correspondence to: Eleanor Spens <eleanor.spens.20@ucl.ac.uk>.

Presented at the Large Language Models and Cognition Workshop at ICML 2024.

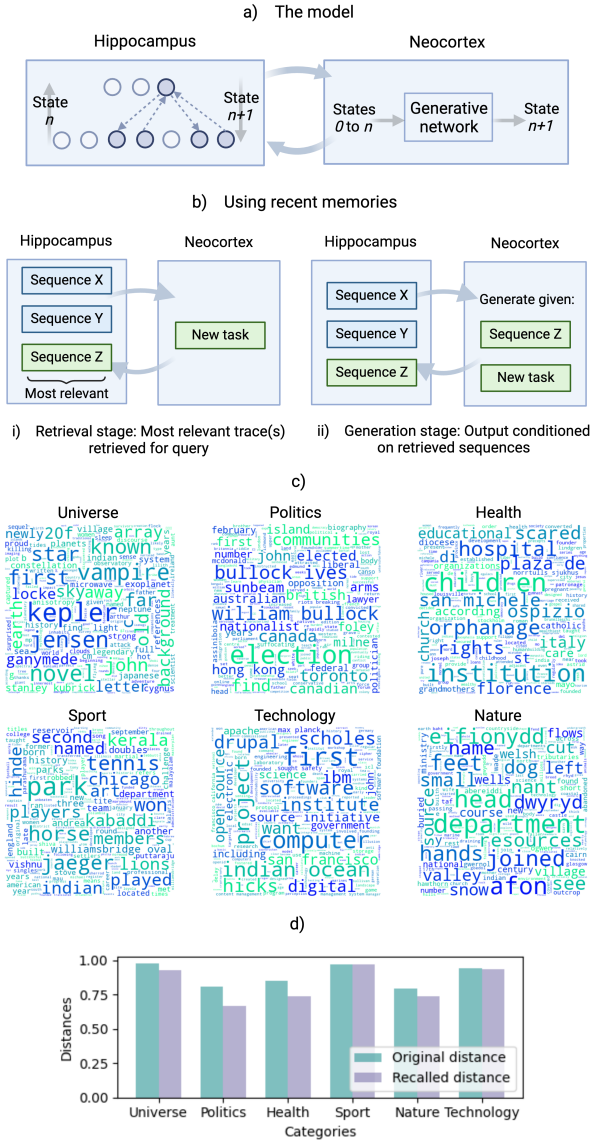

Figure 1. Summary of model. a) Episodic memories are first encoded in the hippocampus, then replayed during rest to train a generative network (GPT-2). b) Even prior to consolidation, the neocortical and hippocampal networks can work together to solve problems with 'retrieval augmented generation' (Lewis et al., 2020). c) When the Bartlett (1932) story is 'consolidated', it is distorted towards the 'priors' of the neocortical model. Word clouds show semantic intrusions in the recalled stories (sampled at a temperature of 0.5, and aggregated across five samples) for models pre-trained on different background categories. d) The embeddings of the training data plus the recalled stories are obtained, using the 'all-MiniLM-L12-v2' model from Reimers & Gurevych (2019). The cosine distances between the mean embedding for each category and either the original story (green) or the recalled story (purple) are shown. Recalled stories become more similar to the background dataset. See also Table 1.

replayed sequences from the hippocampus.

3. Results

3.1. Memorisation and 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 story, a Native American myth, was chosen to be culturally unfamiliar, making memory distortions more pronounced. Bartlett found that the story was recalled in a way that was consistent with the students' background knowledge of the world, with confabulation and rationalisation observed, and that memory distortion increased over time (see also Bergman & Roediger, 1999). This simulation aims to test the hypothesis that recalled narratives are distorted based on background semantic knowledge in our model.

To simulate consolidation, we fine-tuned GPT-2 on the Bartlett (1932) story in addition to a 'background dataset'. Recall was tested 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' on recall of narratives, the background dataset was varied, with six categories of article ('Politics', 'Health', 'Universe', 'Sport', 'Nature', and 'Technology') selected from a collection of Wikipedia content (Ziadé, 2024). See SI for further details.

When the Bartlett story is 'consolidated' into the generative network memory distortions are observed, as in the human data (see Table 2, SI). Furthermore, distortions in recalled stories reflect the 'priors' of the generative network. The word clouds in Figure 1c show that new words added to the story (i.e. 'semantic intrusions') are representative of the background dataset used. To quantify this, Figure 1d shows that the recalled stories move closer towards the background dataset in text embedding space. See also Table 1 for selected examples of semantic intrusions from the three models.

3.2. Structural inference

Consolidation not only extracts statistical regularities from episodic memories (Durrant et al., 2011), but also supports 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, as this relies on the common structure of family trees. The relations in these tasks can be seen as edges in graphs, and the Tolman-Eichenbaum machine (TEM; Whittington et al., 2020) simulates this in the domain of multiple tasks with common transition structures.

Table 1. Recalled stories for different models, showing how semantic intrusions reflect the 'priors' of the generative network. Examples are selected from a range of temperatures.

Model	Recalled story
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.

This simulation tests the hypothesis that consolidation enables structural inference in the neocortical model. 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 (one per task) from scratch on random walks on 100,000 graphs and then tested the models' inference abilities on *novel* sequences with the same underlying structure. Figure 2 shows examples of the synthetic training and test data for each task. (Crucially, whilst each graph's structure is the same, each has a unique set of names for the nodes, representing arbitrary features at a particular location.) This was intended to represent - in a very abstract way - sequences of observations that might be experienced, encoded in the hippocampus, then replayed offline. See SI for further details.

To test inference, we defined a set of cycles in the graph for which the final destination could be inferred given the sequence so far (for example, the next node given 'ab EAST cd WEST' can be inferred to be 'ab'). These templates were then populated with random pairs of letters, so that none of the sequences used for testing featured in the training data. Beam search with five beams was used to generate predictions.

Figures 2e and f show the decreasing 'loss' (aggregated error on the training data) of the spatial model and family tree model respectively, indicating improved ability to predict the next node on the set of graphs used for training, which corresponds to the consolidation of previously experienced environments. Figure 2g shows good performance on a range of *novel* structural inference tasks, including surprisingly complex inferences based on up to six 'hops' in the graph. (See Tables 3 and 4 for the full results.)

These results are consistent with the claim that consolidation supports relational inference and generalisation. Furthermore they suggest that models trained on a simple prediction error minimisation objective can learn an abstract transition structure. Unlike in TEM (Whittington et al., 2020), in which structural regularities and arbitrary specifics are factorised by design, the model learns to separate structure and content (i.e. roles in the graph and the entities that fill them). Many inference problems can be framed in terms of graphs or transition structures, so this approach could be more generally applicable.

3.3. Retrieval augmented generation

This simulation aims to test the hypothesis that the generative network and hippocampal network could work together to support problem solving immediately after encoding, in a way resembling 'retrieval augmented generation' (RAG). Inference from recent memories is modelled as a process whereby relevant sequences from the hippocampus are retrieved and used to condition the generative model.

We created a 'toy example' using the two models from the structural inference results above. 100 new graphs were constructed, each missing one edge, so that inference from memory could be tested. A walk on each of these graphs was stored in the 'hippocampus' (simply a list of strings in this simulation). Crucially each walk contained sufficient information from which the missing edge could be inferred. For each missing edge, a corresponding query was constructed (e.g. if the 'cd PARENT_OF ef' edge was omitted from the graph, the test would be the model's continuation from 'cd PARENT_OF').

Testing involves two stages, retrieval followed by generation (see Figure 1b): 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. (Beam search with five beams was used to generate predictions.) The results show that a RAG-inspired system 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 3).

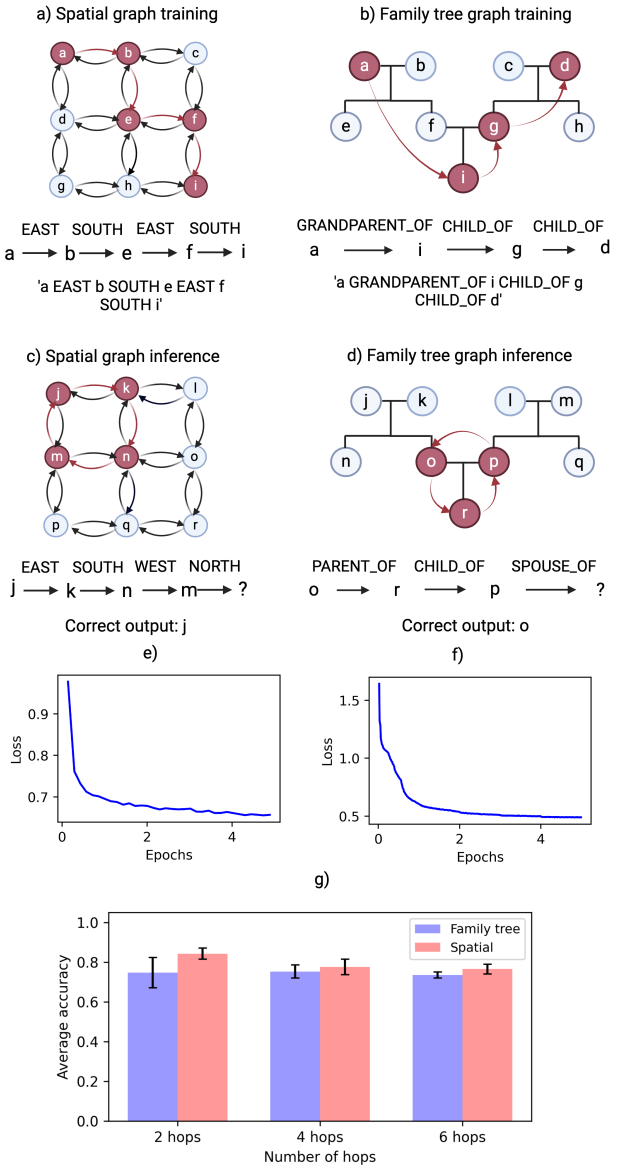

Figure 2. Modelling structural inference. a) Training data for the spatial task consists of trajectories in simplified environments with a shared 3x3 grid structure, but different 'observations' at each node. b) Training data for the family tree task consists of walks on family tree graphs with a shared structure, but different 'names' at each node. c) Novel spatial inference example. d) Novel family tree inference example. e) Loss for GPT-2 trained from scratch on spatial dataset. f) Loss for GPT-2 trained from scratch on family tree dataset. 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 3 and 4 for the accuracies for each template.

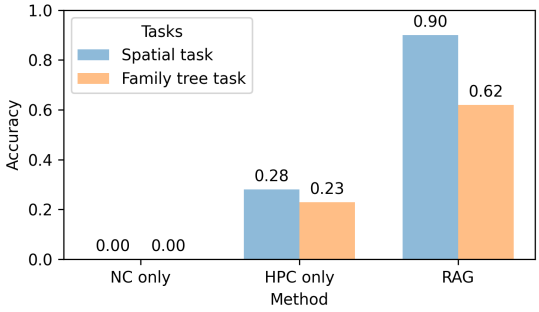

Figure 3. Inference based on recent memories: comparing retrieval augmented generation (RAG) to two baselines. In RAG, the most relevant sequence from the hippocampus is used to condition the generative network. The 'hippocampus only' baseline randomly selects one of the locations / people in the retrieved sequence. The 'neocortex only' baseline conditions the generative network on the task alone, without retrieving sequences from the hippocampus. (The models trained in Section 3.2 are used as the generative networks.)

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 transition probabilities between states through prediction error minimisation. This network exhibits a number of capabilities in addition to the memorisation of 'replayed' sequences, such as spatial and relational inference (Section 3.2). The computational approach taken is applicable to any sequence of symbols, meaning that linguistic and non-linguistic sequences can be modelled in a consistent way: we also demonstrated how distortions arise in narratives, and how these reflect priors in the generative model (Section 3.1). We also suggest that retrieval augmented generation (Lewis et al., 2020) is a potential model of how the generative network might interact with stored sequences in the hippocampus, with relevant memories retrieved from the hippocampus to condition the generative network (Section 3.3).

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, often by prediction error minimisation (Friston, 2010; Káli & Dayan, 2000; 2002). Similarly, there has been a transition in machine learning from task-specific models to larger task-general ones, sometimes referred to as 'foundation models' (Bommasani et al., 2021). We should perhaps think of the brain as learning neural 'foundation models' too, and this paper and others suggest how memory consolidation could contribute to their development.

References

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

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

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. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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

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

Durrant, S. J., Taylor, C., Cairney, S., and Lewis, P. A. Sleep-dependent consolidation of statistical learning. *Neuropsychologia*, 49(5):1322–1331, 2011.

Ellenbogen, J. M., Hu, P. T., Payne, J. D., Titone, D., and Walker, M. P. Human relational memory requires time and sleep. *Proceedings of the National Academy of Sciences*, 104(18):7723–7728, 2007.

Fayyaz, Z., Altamimi, A., Zoellner, C., Klein, N., Wolf, O. T., Cheng, S., and Wiskott, L. A model of semantic completion in generative episodic memory. *Neural Computation*, 34(9):1841–1870, 2022.

Friston, K. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

Hemmer, P. and Steyvers, M. A Bayesian account of reconstructive memory. *Topics in Cognitive Science*, 1(1): 189–202, 2009.

Káli, S. and Dayan, P. Hippocampally-dependent consolidation in a hierarchical model of neocortex. *Advances in Neural Information Processing Systems*, 13, 2000.

Káli, S. and Dayan, P. Replay, repair and consolidation. *Advances in Neural Information Processing Systems*, 15, 2002.

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

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

Norman, Y., Raccah, O., Liu, S., Parvizi, J., and Malach, R. Hippocampal ripples and their coordinated dialogue with the default mode network during recent and remote recollection. *Neuron*, 109(17):2767–2780, 2021.

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

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

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

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

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

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

Ziadé, T. Wikipedia topics. https://huggingface.co/datasets/tarekziade/wikipedia-topics, 2024. Accessed: 2024-05-31.

A. Supplementary information

A.1. Modelling sequence learning

The primary goal during the training of GPT-2 (Radford et al., 2019) 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. 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, and one can either 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:

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

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), and the default GPT-2 tokeniser is used.

As described above, the objective for training is causal language modelling, the task of predicting the next token ('chunk' of characters) in sequences from the training data. This is achieved with the Transformers Python library (Wolf et al., 2019). 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.

A.2. Sampling options

There are many ways to generate sequences given a trained sequence model like GPT-2. As a reminder, a token is a group of commonly co-occurring characters. The same tokeniser is used as in the pre-trained GPT-2 model (Radford et al., 2019).

Greedy decoding, where the model selects the token with the highest probability as the next token in the sequence, is the simplest way to generate sequences. However this can lead to repetitive and predictable sequences, as greedy decoding always opts for the most likely option without exploring potential alternatives. 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'. 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.

To be more precise, the equation below describes the 'softmax with temperature' function that is applied to the vector of scores for each token. The softmax function transforms this vector of scores into a vector of *probabilities*. As the numerator is an exponential, 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. The denominator normalises each element in the vector by the sum of all the exponentials, ensuring the probabilities add up to one:

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

Top-K sampling limits the model's choice to the K most likely next words and samples from this subset according to their probability distribution. This prevents the model from picking highly improbable words, reducing the risk of generating nonsensical text. Unlike top-K sampling, top-p (nucleus) sampling uses a cumulative probability threshold (p) and then selects from the smallest set of items whose combined probability is below this threshold. This method allows the model to consider a broader or narrower set of options depending on the certainty of its predictions, which can lead to more coherent outputs. Beam search is not a sampling method but a search strategy that expands on greedy decoding by considering 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. Beam search is particularly useful for tasks requiring high-quality outputs, such as translation or summarisation, but can be computationally intensive.

Table 2. Recalled stories for different temperatures, for a model trained on the Bartlett (1932) story plus the 'Sport' category of the Wikipedia dataset (Ziadé, 2024) as the background data distribution. The continuation given 'One night two young men from Egulac' is shown.

Type	**Recalled story**
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: ...
No sampling	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 ...

A.3. Modelling distortions in narratives

To simulate consolidation, the medium-sized GPT-2 model was fine-tuned on the Bartlett (1932) story in addition to items from one of six 'background datasets', allowing the network's 'priors' to be manipulated based on the choice of dataset. Recall was tested by inspecting the continuation of the story at a given temperature. A dataset of Wikipedia content (Ziadé, 2024) was used, with six categories of article selected to provide a wide range of style and content ('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 (1932) story. Each model was trained for 50 epochs on this combined dataset.

Figure 1c shows 'semantic intrusions' at a temperature of 0.5, aggregated across ten sampled 'memories' of the story. Word clouds, created with the 'wordcloud' Python package, are used to visualise these intrusions. They show terms in the recalled Bartlett stories which did not appear in the original (with common words, i.e. 'stopwords' like 'the', excluded). Table 2

shows the effect of varying the temperature. Recall at a higher temperature becomes more 'imaginative', but even without sampling distortions are observed.

To show more quantitatively that recalled stories are distorted towards the 'background dataset', the 'all-MiniLM-L12-v2' model from Reimers & Gurevych (2019) was used to obtain the embeddings of the training data, plus those of the original and recalled stories. The cosine distances between the mean embedding for each category and either the original story or the recalled story are shown in Figure 1d. (Specifically, five recalled stories were sampled at a temperature of 0.1, and the average distance to the category mean was calculated to give the purple bars in this figure.)

A.4. Modelling structural inference

In the spatial task, 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 2a). Whilst each graph's structure is the same, nodes are labelled with names to represent arbitrary features at a particular location (random pairs of letters are used to increase the possible number of names). 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'. The family tree graph has a simple structure for illustrative purposes, consisting of two children, their parents, and two sets of grandparents. See Figure 2b. We model this as a directed graph with edges for different relationships. As in the spatial graph 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'.

In each case, we created 100,000 graphs with the same structure but randomly chosen values (pairs of letters) for the nodes. A random walk of 50 transitions was sampled from each graph to create the training data, which represent sequences of observations that might be experienced, encoded in the hippocampus, then replayed offline. (These random walks were not filtered to cycles, meaning that the models' inference abilities could not be attributed to learning that all sequences start and end at the same node.) GPT-2's medium-sized architecture was then trained from scratch for five epochs.

After training the models, we tested novel inference. We chose 'loops' in the graph for which the final destination could be inferred given the sequence so far (for example, the next node given 'ab EAST cd WEST' can be inferred to be 'ab'). The templates were then populated with random pairs of letters, so that each sequence was novel, before testing the accuracy of inferring the final node. Note that only the next node prediction was used to evaluate inference but predictions continued beyond this point (i.e. intersecting the path in the prompt). Tables 3 and 4 give the average score for each 'template', while Figure 2g aggregates these results by the number of graph transitions in the sequence (or 'hops'). Many other tests could be performed, e.g. of the ability to generate structurally valid graphs given a prompt.

Table 3. Family tree task inference templates and their average accuracies.

Inference template	**Average accuracy**
{} CHILD_OF {} PARENT_OF {}	0.81
{} PARENT_OF {} CHILD_OF {}	0.76
{} GRANDCHILD_OF {} GRANDPARENT_OF {}	0.80
{} GRANDPARENT_OF {} GRANDCHILD_OF {}	0.62
{} CHILD_OF {} CHILD_OF {} GRANDPARENT_OF {} SIBLING_OF {}	0.70
{} 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 4. Spatial task inference templates and their average accuracies.

Inference template	**Average 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 {}	0.80
{} NORTH {} EAST {} SOUTH {} WEST {}	0.81
{} EAST {} EAST {} NORTH {} WEST {} WEST {} SOUTH {}	0.79
{} NORTH {} NORTH {} WEST {} SOUTH {} SOUTH {} EAST {}	0.74