bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

A neural network model of free recall learns multiple memory strategies

Moufan Li[1], Kristopher T. Jensen[2], Qiong Zhang[3], Qihong Lu[4,5], and Marcelo G. Mattar[1]

[1] Department of Psychology, New York University
[2] Sainsbury Wellcome Centre, University College London
[3] Department of Psychology, Rutgers University
[4] The Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University
[5] Center for Theoretical Neuroscience, Columbia University

Abstract

Humans exhibit structured patterns of memory recall, including a tendency to recall more recent information and to recall events in the same order they were experienced. Classic computational models explain these patterns by positing that memories incorporate the ongoing "temporal context", formed by smoothly integrating the stimulus history. However, it is unclear whether a single mechanism can account for the full repertoire of human memory strategies, as the optimal approach may be task-dependent. For example, human memory experts widely apply the "memory palace" strategy, which is empirically better but not captured by temporal context models. Here we show that neural networks optimized for free recall develop diverse retrieval strategies, with only some of them resembling temporal context models. The best-performing models discovered a stimulus-invariant index code that emphasizes the studied position of each list item, instead of its temporal context. This creates a stable scaffold for forward recall akin to the memory palace technique. This index code was more likely to emerge when networks were i) encouraged to recall all studied items rather than prioritizing a few items, and ii) prevented from relying on recency, resonating with human data. Our findings demonstrate that human-like recall patterns can arise from multiple distinct computational mechanisms, and that sequential retrieval using item index is an optimal strategy that explains expert-level recall performance.

1 Introduction

Decades of research into episodic memory have sought to understand how humans search their past experiences to recall information (Tulving and Thomson, 1973; Johnson and Rugg, 2007; Polyn and Kahana, 2008). The free recall task, where participants study a list of items and then recall them in any order, has become a central paradigm for this investigation. Human recall in this paradigm consistently reveals highly structured patterns, including serial position effects of enhanced memory for the first and last items (Murdock Jr, 1962), and the temporal contiguity effect, describing people's tendency to recall memories in nearby list positions to the just recalled memory and in the same order as they were studied (Kahana, 1996; Howard and Kahana, 1999). These patterns, replicated and extended over decades of research (Glenberg et al., 1980; Koppenaal and Glanzer, 1990; Zaromb et al., 2006; Polyn et al., 2009b; Healey et al., 2019), provide strong constraints on theories of how episodic memory organizes and retrieves information. Yet, these general patterns coexist with striking variability in individual memory ability and strategy (Healey et al., 2014; Unsworth, 2019). This raises a crucial question: what computational mechanisms can account for this diversity, and under what circumstances is a particular strategy optimal?

A prominent class of computational models explains these memory patterns by positing the existence of a gradually changing internal state, or temporal context, that evolves with each experienced item and becomes associated with them in memory. The most influential of these, the Temporal Context Model (TCM), proposes that this context is reinstated during memory recall and serves as a dynamic cue to retrieve

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

subsequent memories (Howard and Kahana, 2002; Sederberg et al., 2008; Polyn et al., 2009a). This simple mechanism explains many recall phenomena and has become the standard model of free recall. However, it is possible that alternative computational strategies, distinct from a drifting temporal context, could also support effective recall performance and account for the observed recall patterns.

One important alternative computational strategy for free recall is the "memory palace" (method of loci), a technique widely used by memory experts to achieve extraordinary recall performance (Roediger, 1980; Wilding and Valentine, 1997). The memory palace strategy involves mentally placing items along a sequence of familiar locations during encoding and systematically retracing these locations during retrieval (Maguire et al., 2003; Dresler et al., 2017; Wagner et al., 2021). Computationally, this approach is fundamentally different from TCM. Instead of relying on a drifting, content-based context, the memory palace creates a stable scaffold by representing each item's serial position independently of its content—a mechanism more akin to positional coding models (Neath and Crowder, 1990; Brown et al., 2000; Logan and Cox, 2021). Although rational analyses (Anderson and Milson, 1989; Lieder and Griffiths, 2020) suggest that such forward-ordered recall is indeed an optimal policy for free recall (Zhang et al., 2023), this work has been constrained by the architectural assumptions of TCM. Critically, these assumptions cannot account for the stable, "index"-based strategy of the memory palace, leaving it unclear how to computationally explain expert-level performance and what alternative optimal strategies might exist. We therefore ask what computational strategy does a system trained to maximize performance converge to. Does it rely on a drifting, content-based temporal context, or would does it instead discover a different mechanism, such as an index-based strategy?

Here, we use a neural network to discover optimal memory strategies empirically (Richards et al., 2019), an approach which provides more flexibility for context updating compared to hand-crafted cognitive models, such as TCM. We designed a model composed of a recurrent neural network (RNN) augmented with an external episodic memory buffer (Lu et al., 2022; Whittington et al., 2020) and trained it using reinforcement learning. The model was optimized with the sole objective of maximizing the number of recalled items without any bias on the order of recall. By analyzing the internal representations and behaviors that emerge from this optimization process, we can identify the computational principles that govern effective memory search, thus offering a complementary perspective on the diversity of strategies that can facilitate free recall.

Our results reveal that neural networks trained solely to maximize recall performance could develop various computational strategies, many of which exhibiting human-like recall patterns such as the temporal contiguity effect. Remarkably, while some networks converged on TCM-like mechanisms based on drifting temporal context, other networks discovered different strategies. A particularly common strategy involved representing each item's abstract serial position independently of content and systematically traversing these index representations during retrieval to recall items in strict forward order — a strategy functionally analogous to the memory palace technique. This index-based strategy achieved superior performance and robustness compared to TCM-like mechanisms, providing a normative account for why perfect forward-ordered recall is so effective and supporting theoretical claims about its optimality (Zhang et al., 2023). These findings establish that multiple computational strategies can produce human-like recall, and that learning an item-independent scaffold for memory organization leads to a powerful, and potentially optimal, solution for free recall.

2 Results

2.1 A neural network model of sequential episodic memory

To investigate the computational principles of memory retrieval, we simulated the classic free recall task (Figure 1A). This task consists of a *study* phase where a list of items is presented sequentially, followed by a *response* phase where the model must retrieve as many items as possible in any order. We designed a neural network model capable of encoding and retrieving sequential memories (Figure 1B-C). The model consisted of a "context module" that uses gated recurrent units (GRUs) (Cho et al., 2014) as working memory, and a "memory module" with slots for storing episodic memories (Ritter et al., 2018; Lu et al., 2022). This

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

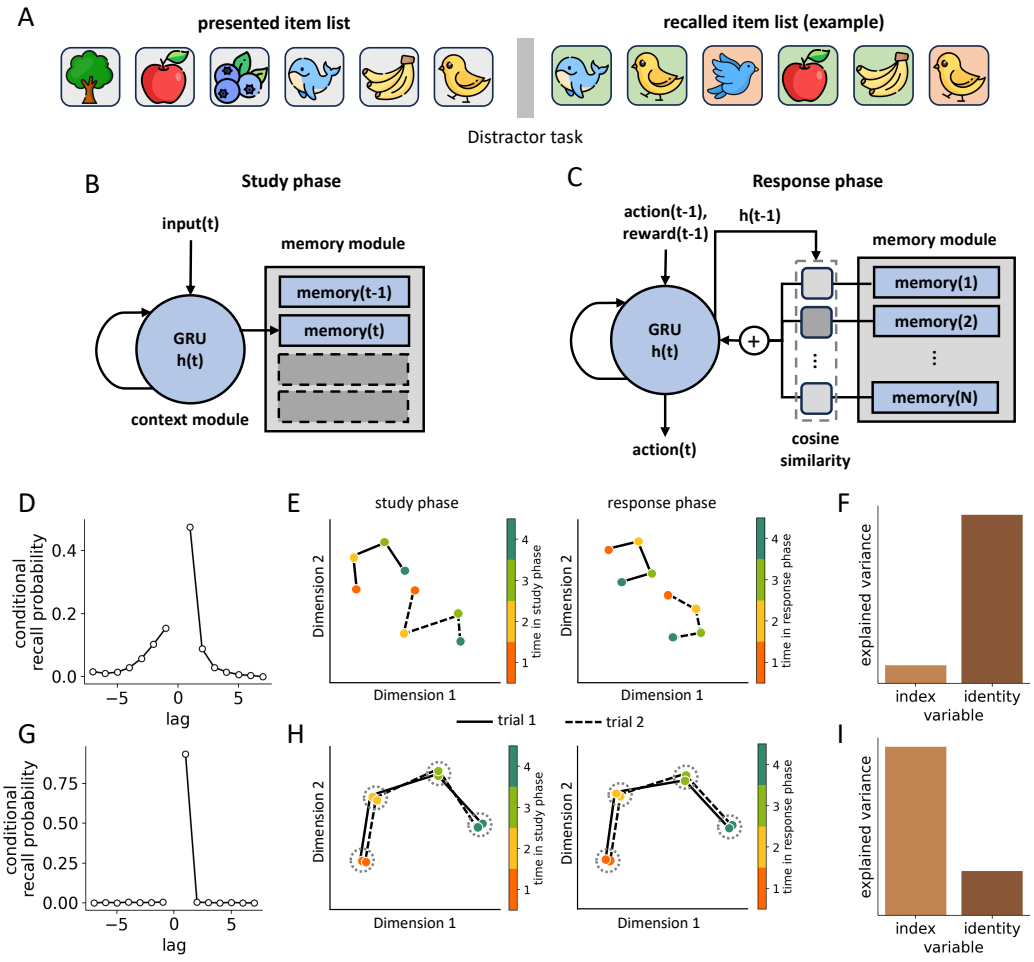


Figure 1: **Task and model schematic.** (A) Free recall task. The model receives a one-hot encoded item (shown as icons) at each time step during the study phase. In the response phase, it receives a reward for each correct recall (green) and a penalty for each incorrect or repeated recall (red). (B–C) Model architecture. A GRU context module is connected to a memory module. (B) Study phase: each input item updates the hidden state, which is then appended to the memory module. (C) Response phase: the model computes cosine similarity between the current hidden state and stored memories, retrieving the most similar one. The hidden state is then updated as a function of the previous hidden state, current input, and retrieved memory. Previous action and reward are also provided as inputs during this phase. (D–F) Conceptual illustration of the Temporal Context Model (TCM). (D) TCM predicts a smooth conditional recall probability (CRP) curve: items studied close in time to the last recalled item are more likely to be recalled next, with a forward-order bias. (E) Evolution of the TCM internal state (context) during study (left) and response (right). The state drifts toward each encoded or retrieved item, producing distinct trajectories across trials; recall order can therefore differ from encoding order. (F) Variance in the internal state explained by item index versus item identity in TCM. (G–I) Conceptual illustration of the memory palace technique. (G) The memory palace predicts perfect forward recall. (H) Evolution of the internal state in the memory palace. The state represents item index largely independently of item identity, following a fixed trajectory across trials in both study and response phases. (I) Variance in the internal state explained by item index versus identity in the memory palace.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

architecture, while analogous to the context and feature spaces in the Temporal Context Model (TCM) (Howard and Kahana, 2002), is intentionally flexible to allow for the emergence of novel context-updating mechanisms.

Each trial in the task consisted of a list of 8 items randomly sampled from a dictionary of 64 items, represented as one-hot vectors. During the task's *study phase*, the model sequentially processes a list of items and stores its internal GRU state in the memory module after each item (Figure 1B). In the subsequent *response phase*, the model uses its current GRU state as a retrieval cue to find the most similar memory in the memory module, recalls the associated item, and updates its state accordingly (Figure 1C). Crucially, to simulate the distractor conditions of human experiments and force reliance on the episodic memory module (Glanzer and Cunitz, 1966), we reset the network's hidden state between the study and response phases.

To discover effective recall strategies without imposing prior constraints, we trained the model using the advantage actor-critic (A2C) reinforcement learning algorithm (Mnih et al., 2016; Jensen, 2023). The model was rewarded simply for maximizing the number of correctly recalled items and penalized for errors, with no explicit instruction on the order of recall. To determine if any emergent strategies resembled human behavior, we tested whether the model would spontaneously reproduce the temporal contiguity effect—the human tendency to recall items that were studied in close temporal proximity to the just recalled item, and in the forward order (Kahana, 1996). This signature pattern of human recall can be visualized with a conditional recall probability (CRP) curve, which plots the likelihood of recalling an item as a function of its proximity (lag) to the previously recalled item (Figure 1D).

We hypothesized that high recall performance could be achieved through at least two qualitatively different strategies. First, the network might converge on a mechanism based on a drifting temporal context analogous to the Temporal Context Model, using a drifting, item-dependent temporal context to guide recall (Figure 1D-F). This mechanism would be characterized by a smooth CRP curve (Figure 1D), internal states that follow distinct item-dependent trajectories on each trial (Figure 1E), and internal representations dominated by item identity (Figure 1F). Alternatively, the network might discover a more structured, "memory palace"-like strategy that leverages a stable, item-independent scaffold to represent each item's serial position (Figure 1G-I). This mechanism would be characterized by rigid, forward-ordered recall (Figure 1G), stereotyped neural trajectories across trials (Figure 1H), and internal representations dominated by item index (Figure 1I). Distinguishing between these possibilities would therefore reveal the computational principles underlying the model's emergent solution to the free recall task.

2.2 Multiple strategies emerge in a neural network model optimized for free recall

We first asked what computational strategies would emerge when an RNN was given full flexibility to maximize recall performance. After optimizing a diverse population of networks with varying random seeds and hyperparameters, we observed that the networks did not converge on a single solution. Rather, we observed a striking diversity in both behavioral patterns and internal representations. To formally categorize these solutions, we performed k-means clustering on the models using metrics that captured both their recall behavior (forward asymmetry, the model's tendency to recall in a forward order, and temporal factor, the extent of temporal contiguity) and their internal representations (variance explained by index vs. identity). This analysis revealed the existence of three qualitatively distinct and well-separated strategies for solving the free recall task (Figure 2A). These strategies were distinguished by both their recall behavior—the trade-off between forward asymmetry and temporal contiguity (Figure 2B)—and their internal representations—the balance between coding for an item's index versus its identity (Figure 2C). Despite their differences, all three strategies achieved high performance (Figure 2D).

To understand the nature of the most dominant high-performance strategy, we analyzed its functional properties. Models employing this strategy exhibited near-perfect, forward-ordered recall, producing a sharp conditional recall probability curve (Figure 2E). Critically, their internal states followed a consistent, stereotyped trajectory during both study and retrieval, regardless of the specific items being processed (Figure 2F). Further analysis showed that an item's serial index explained substantially more variance in the hidden

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

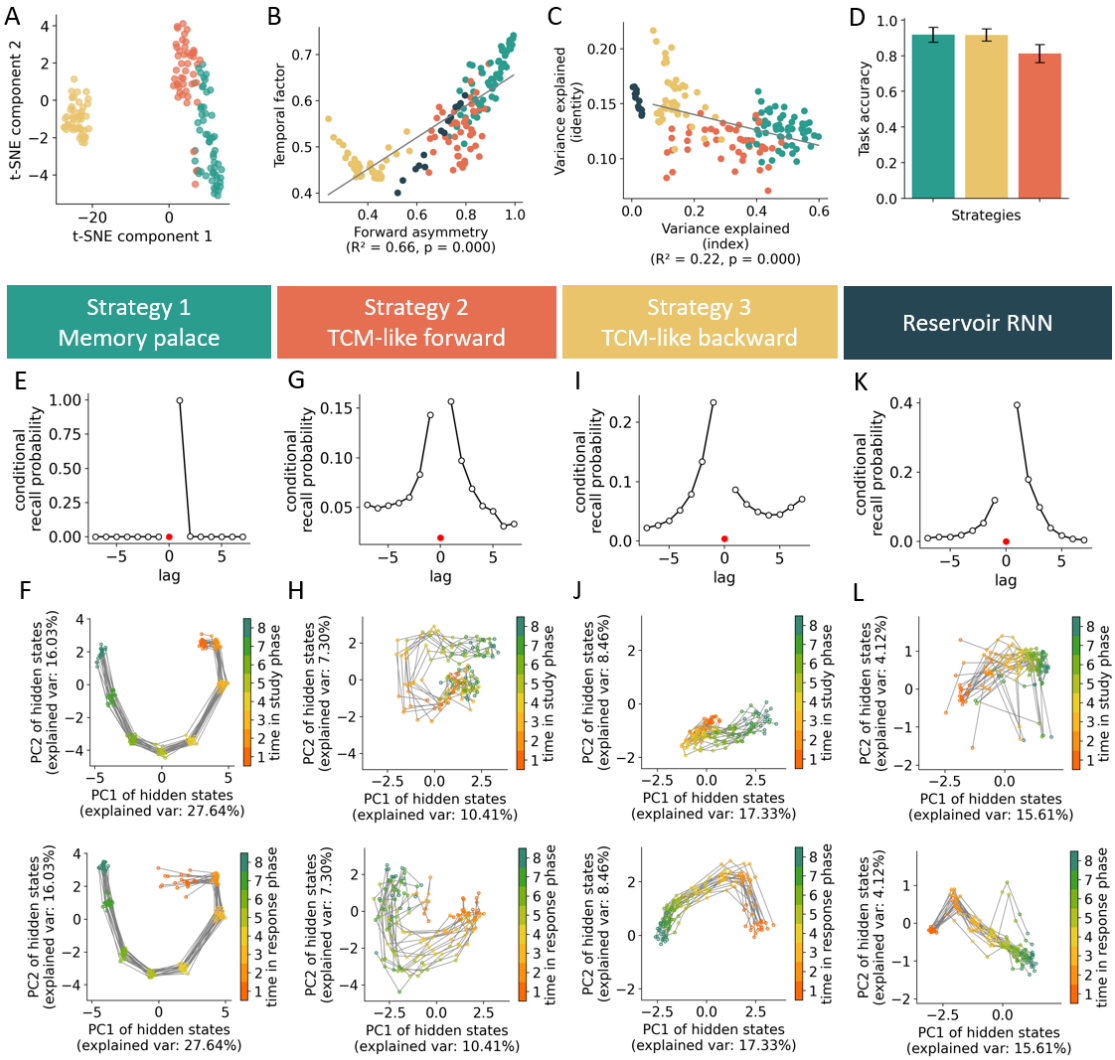


Figure 2: **Different strategies of free recall.** (A) tSNE embedding of all trained networks based on clustering metrics. Colors denote cluster identity. (B) Relationship between forward asymmetry and temporal factor. Colors indicate strategy groups; dots show reservoir RNN samples. The black line is a linear regression fit across trained RNNs (excluding the reservoir). Forward asymmetry greater than 0.5 reflects a forward-recall bias, while higher temporal factor corresponds to a steeper conditional recall probability (CRP) curve. (C) Relationship between hidden state variance explained by item index and hidden state variance explained by item identity. (D) Mean task performance for each strategy (error bars: standard deviation). (E–L) Representative models for the three emergent strategies and the reservoir RNN. (E,G,I,K) CRP curves showing recall probability as a function of lag relative to the previously recalled item. (F,H,J,L) The first two principal components (PCs) of hidden states during study (top) and response (bottom) phases. Colored dots represent hidden states at each time step; trajectories indicate different trials. A single PCA was fit jointly to study and response activity for each model. Each plot shows trajectories from 10 trials with distinct input lists. In the study phase, hidden states are taken immediately after each input. In the response phase, hidden states are taken immediately before memory retrieval (i.e., the state determining which memory is recalled).

states than its identity (Figure 3A). Together, these features indicate that the network had learned a stable, item-independent scaffold for memory organization, a mechanism functionally analogous to the "memory palace" technique.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

We next examined the other two emergent strategies to see how they differed. These strategies closely resembled the mechanisms of the Temporal Context Model (TCM), showing smooth temporal contiguity effects but with opposing directional preferences. The "TCM-like forward" strategy showed a forward recall bias (Figure 2G) and greater trial-to-trial variability in its neural trajectories (Figure 2H). In contrast, the "TCM-like backward" strategy preferred to recall items in reverse order (Figure 2I-J). For both of these strategies, item identity explained more variance than item index (Figure 3A). Thus, our model could not only discover the novel memory palace solution but also learned strategies that operate on principles aligned with existing cognitive models.

To establish a benchmark for these learned solutions, we tested whether recurrent networks would reproduce the temporal context mechanisms of existing cognitive models without learning. We created a "reservoir" network by freezing the random recurrent weights of our model, thereby preventing it from optimizing its dynamics for the task, and trained only its input and output weights. This network developed a strategy that resembled the Temporal Context Model (TCM), exhibiting a classic temporal contiguity effect with forward asymmetry (Figure 2K). An analysis of its internal state revealed that it differed in each trial depending on the specific items presented (Figure 2L), and that it encoded significantly more information about item identity than item index (Figure 3A). This result demonstrates that a TCM-like solution is the default behavior for this architecture, confirming that the memory palace is a more specialized strategy discovered through end-to-end optimization.

A critical question was whether the index-based memory palace strategy is fundamentally incompatible with the smoothly drifting dynamics of TCM. To answer that question, we first decoded item identity information from the memory palace network's hidden states and found that it contained a clear, gradually decaying representation of the currently processed item (Figure S1A,B). Furthermore, an analysis of the hidden states' cosine similarity revealed smooth temporal dynamics across all strategies, with adjacent states in time being more similar (Figure S1C,D). These findings demonstrate that memory palace and temporal context mechanisms are not mutually exclusive; the network can maintain a stable positional scaffold while simultaneously encoding smoothly drifting content information.

For the memory palace strategy to be effective, its positional code must act as a stable scaffold, remaining invariant across different items (item-invariance) and consistent between encoding and retrieval (phase-invariance). To test this, we evaluated our memory palace models using cross-condition generalization performance (CCGP) (Bernardi et al., 2020). Specifically, we trained a decoder to classify item index from hidden states during the study phase, using trials generated from only 8 random items, and then tested the decoder on hidden states during the response phase with trials containing all 64 items. Models employing the memory palace strategy achieved high cross-decoding accuracy of index, indicating that the same positional code supported both encoding and retrieval across different item sets (Figure 3B). In contrast, reservoir and TCM-like models showed poor generalization of index representations. This confirms that the memory palace relies on an item-independent index code that can be reinstated during retrieval, analogous to how human experts retrace fixed spatial locations when applying the method of loci. We also assessed the cross-decoding accuracy of identity by training a decoder on hidden states at a single study-phase time step and testing on all time steps in the response phase. Here, all strategies—including memory palace, TCM-like, and reservoir models—showed relatively high cross-decoding accuracy of identity (Figure 3C).

Finally, we sought to determine if a model's reliance on an index-based code was directly related to its tendency for forward-ordered recall. Across the entire population of trained models, we found a strong positive correlation between the variance explained by index and behavioral metrics of forward recall and temporal contiguity (Figure 3D,E). This result establishes a direct link between the underlying computational mechanism and the resulting recall behavior, showing that models that developed a stronger item-independent scaffold were precisely those that adopted a more systematic, forward-ordered recall policy.

2.3 The impact of training conditions on the learned strategies

Having identified multiple distinct strategies in the RNNs trained on free recall, we next sought to understand the conditions under which each strategy emerges. First, we asked how prioritizing long-term rewards over

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

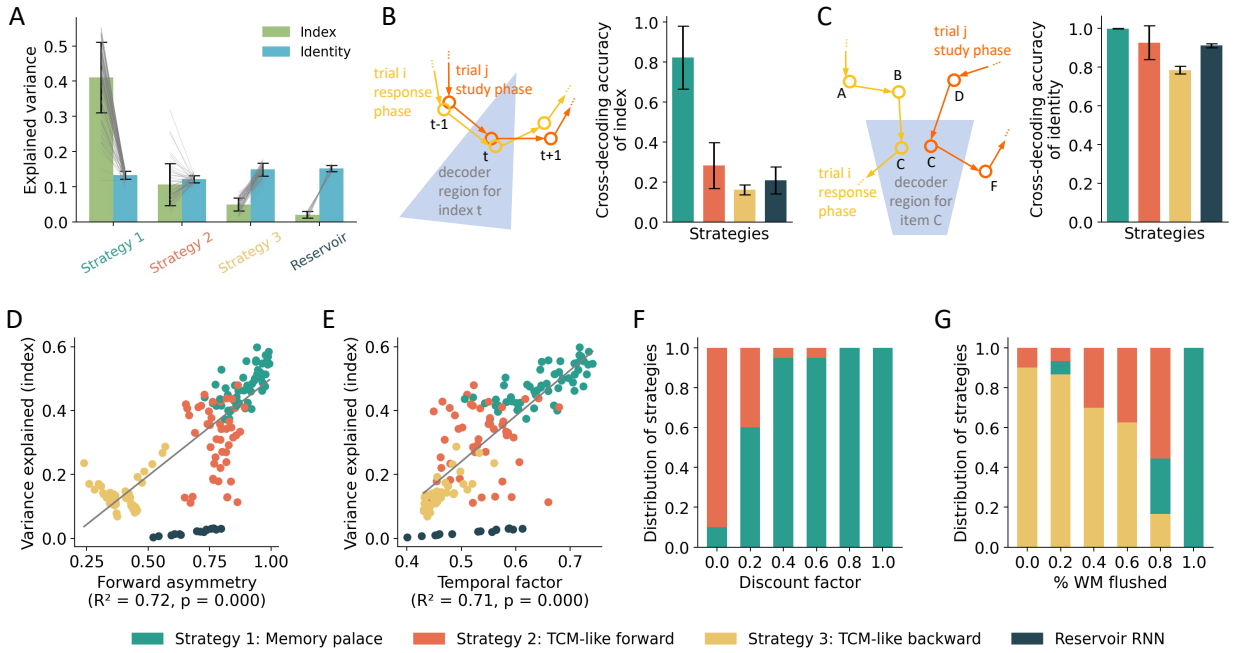


Figure 3: **Model representations and factors influencing strategy.** (A) Variance in hidden states explained by item index and identity for RNNs with each strategy and the reservoir network. Bars show the average across models, error bars show standard deviation, and gray lines show individual models. (B) Cross-decoding accuracy of item index. A decoder trained on hidden states from study-phase trials with only 8 unique items was tested on response-phase trials with all 64 items, and vice versa. (C) Cross-decoding accuracy of item identity. A decoder trained on hidden states from a single study-phase time step was tested across all response-phase time steps, and vice versa. (D–E) Relationship between recall behavior and representational coding. (D) Forward asymmetry versus variance explained by item index. (E) Temporal factor versus variance explained by item index. Gray lines indicate linear regression fits. (F–G) Training conditions influence strategy distribution. (F) Proportion of models adopting each strategy across different discount factors (20 models per condition; working memory flush held constant). (G) Proportion of models adopting each strategy across different levels of working memory flush (20 models per condition; discount factor held constant).

immediate ones would influence the learned strategy. We manipulated this by varying the reinforcement learning discount factor, γ, where a high value encourages the model to optimize for recalling the entire list rather than just the next item. We found that a higher discount factor led to a significantly greater proportion of models learning the memory palace strategy over the TCM-like forward strategy (Figure 3F). This result aligns with findings from human studies, where instructions to recall a whole list encourage participants to initiate recall at the beginning of the list and proceed in a forward order (Tan et al., 2016). Thus, an optimization process that prioritizes long-term, global rewards is a key factor in developing the memory palace strategy.

Second, we asked whether the initial state of working memory at the start of recall would affect the emergent strategy. To simulate the effect of a distractor task—commonly used in human free recall experiments—, we modulated the amount of working memory being "flushed" by injecting a controlled amount of noise into the hidden state between the study and response phases. When working memory was fully preserved (0% flush), the model was biased by the most recent items and consistently adopted the TCM-like backward strategy (Figure 3G). Conversely, when working memory was entirely flushed (100% flush, effectively resetting the hidden state to a random vector), this recency bias was eliminated, and the model overwhelmingly learned the memory palace strategy, initiating recall from the beginning of the list. This suggests that the initial state of working memory is a critical determinant of recall strategy, potentially explaining how humans can

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

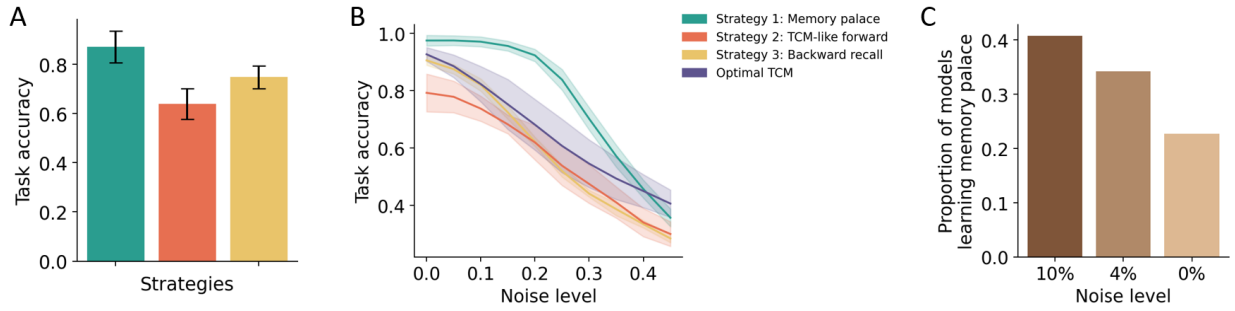


Figure 4: **Performance and noise robustness of different strategies.** (A) Mean task performance of models within each strategy for a total of 100 models with the same training condition (discount factor = 1.0, proportion of working memory flushed = 0.9). Models were assigned to clusters based on the same cluster centers as in figure 2 (B) Mean task performance of each strategy as a function of the proportion of noise injected into the recurrent hidden state, together with an optimal TCM model. (C) Proportion of models that learn a memory palace strategy (variance explained by index larger than 40%) depending on the proportion of noise injected to the RNN hidden state during training.

improve performance by actively resetting their mental state to overcome the influence of recent events. These results are also consistent with rational analyses of TCM showing that forward-ordered recall is only optimal when retrieval begins from the start of the list (Zhang et al., 2023).

2.4 The memory palace strategy is an optimal and robust solution

Having established that our model can learn multiple strategies, we next sought to determine which of them represents the most effective solution for free recall. To ensure a fair comparison, we trained a new set of models with different random seeds under identical conditions and classified their strategies using the cluster centers defined earlier. When comparing performance, the memory palace strategy consistently outperformed both TCM-like forward and backward strategies (Figure 4A). Moreover, task performance was positively correlated with forward asymmetry, temporal factor, variance explained by index, and cross-decoding accuracy of index (Figure S3A–D), linking behavioral and representational features of the memory palace to superior recall.

We reasoned that a truly optimal strategy must not only be accurate but also robust, so we next asked how resilient each strategy was to neural noise. We tested this by injecting noise into the hidden states of trained models at test time. This analysis revealed that the memory palace strategy was substantially more resilient to noise than either of the TCM-like strategies (Figure 4B). The memory palace strategy outperformed even an optimized TCM model with perfectly forward recall (Zhang et al., 2023). This suggests that the learned index code provides a stable scaffold for memory retrieval that is inherently more robust than a drifting, content-based context, which may be especially important for longer lists.

Given its superior noise resilience, we then asked whether a noisy environment could actively promote the discovery of the memory palace strategy. We trained new populations of models while adding different levels of noise to their hidden states throughout training. Consistent with our hypothesis, models trained in noisier conditions were significantly more likely to learn the memory palace strategy (Figure 4C). This indicates that the memory palace is not just robust post-hoc, but is preferentially discovered when a network must find a solution that is robust to internal variability.

Taken together, these findings parallel empirical observations in humans, where stronger temporal contiguity and initiating recall from the beginning of a list are associated with better memory performance (Sederberg et al., 2010; Zhang et al., 2023). The superior performance and robustness of the emergent memory palace strategy therefore support the conclusion that this index-based, forward-recall mechanism represents a computationally optimal solution, providing a normative explanation for its effectiveness in both

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

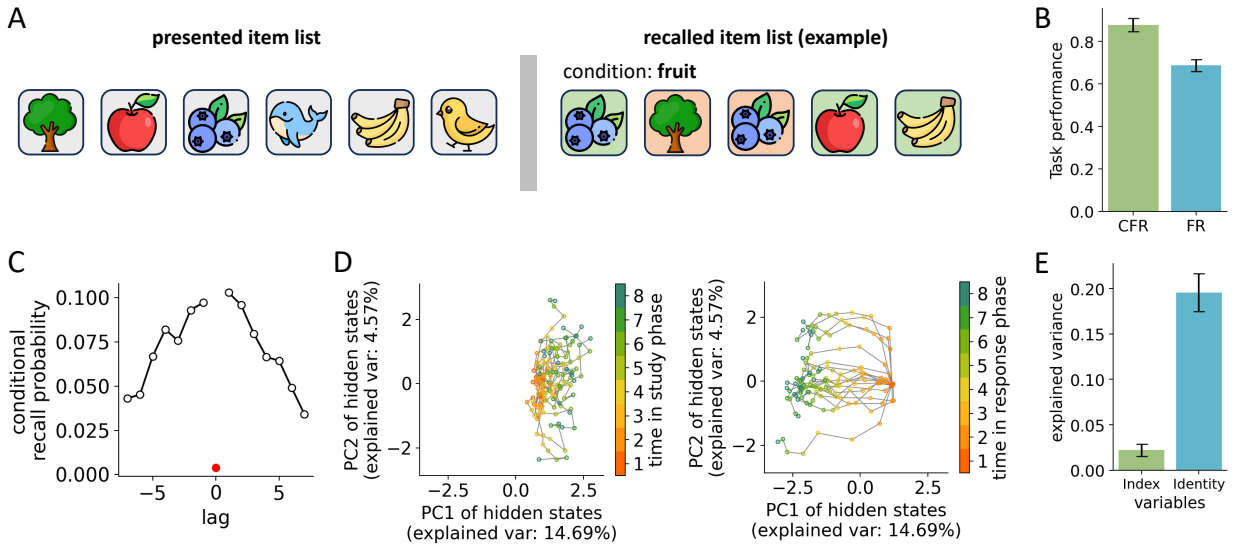


Figure 5: **Content-based memory search discourages the memory palace strategy.** (A) Schematic of the conditional free recall task. Each item was assigned a set of features, and during the response phase the model was cued to recall only items with a specified feature value. Correct recalls (green) were rewarded, while errors (red) occurred if the recalled item was absent from the list, repeated, or failed to match the condition. (B) Task performance of models trained on the conditional recall task and subsequently tested on standard free recall. Bars show the mean across 10 models trained with different random seeds; error bars denote one standard deviation. (C) Conditional recall probability (CRP) curves showing recall likelihood as a function of lag from the previously recalled item. (D) The first two principal components (PCs) of GRU hidden states during the study phase (left) and response phase (right). Each trajectory represents a trial. (E) Fraction of variance in hidden states explained by item index versus item identity.

artificial networks and human memory experts.

2.5 The optimal strategy depends on the task demands

Although memory palace appears highly effective strategy for free recall in both our simulations and human behavior, its scope of optimality remains unclear. We hypothesized that this strategy is particularly well suited to free recall, where success depends only on retrieving labels rather than content. By contrast, everyday memory often requires flexible retrieval based on an item's features or identity, using the content within each memory for downstream tasks. In such settings, we predicted that a purely index-based scaffold would be ill-suited.

To test this hypothesis, we trained models on a "conditional free recall" task that required content-based retrieval. In this task, each item was assigned a set of features, and during the response phase the model was cued to recall only items with a specific feature value (Figure 5A). We trained the model under the same conditions that strongly promoted the memory palace strategy in the free recall task (i.e., a high discount factor and full working memory flush). We then tested this model on a standard free recall task to analyze the learned strategy. We found that these models achieved moderate performance on both the conditional and standard free recall tasks (Figure 5B). Notably, they also exhibited all the hallmarks of a TCM-like strategy: a smooth temporal contiguity effect (Figure 5C), variable trial-to-trial neural trajectories (Figure 5D), and internal representations dominated by item identity rather than index (Figure 5E). These results demonstrate that the optimality of the memory palace strategy is task-dependent, representing a highly specialized solution for sequential recall that is discarded when flexible, content-based retrieval is required.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

3 Discussion

This study demonstrates that a neural network optimized for free recall spontaneously discovers multiple, distinct strategies for solving the task. While some of these strategies were reminiscent of the drifting temporal context assumed by standard cognitive models (Howard and Kahana, 2002; Sederberg et al., 2008), others developed a novel, alternative mechanism: an item-independent scaffold that encodes each item's sequential index to guide systematic forward retrieval. This "memory palace" strategy yielded superior performance and robustness, and its emergence was favored by conditions that prioritized whole-list recall and discouraged reliance on working memory. These results show that human-like recall patterns can arise from different computational principles and suggest that the memory palace represents a normative, computationally optimal solution for free recall.

Our modeling approach provides a normative explanation for the effectiveness of systematic recall strategies, extending previous rational analyses of memory (Zhang et al., 2023). By training a network without built-in biases, we showed that a memory palace strategy emerges not as a contrived mnemonic, but as a high-performing, optimal solution. This finding offers a mechanistic basis for the empirically observed link between strong temporal contiguity and better recall performance in humans (Sederberg et al., 2010), supporting theoretical claims that forward-ordered recall is an optimal policy (Zhang et al., 2023). Our representational analyses further connect this behavior to the formation of an item-invariant index code, akin to the mental locations used by memory experts. Therefore, our results suggest that the memory palace is not merely a contrived mnemonic but may be a general and computationally optimal strategy for free recall.

The factors that promoted the memory palace strategy in our model may also offer testable hypotheses about the origins of individual differences in human recall. We found that a higher reinforcement learning discount factor—the tendency to prioritize future over immediate rewards—was a key determinant of this strategy's emergence. This suggests that individuals more disposed to long-term planning may be more likely to develop systematic, forward-ordered recall strategies, a hypothesis testable in future experiments (Basile and Toplak, 2015; Sadeghiyeh et al., 2020). Similarly, the fact that the index-based strategy was learned most effectively when recency effects were suppressed suggests that the ability to overcome the influence of immediately preceding events is a critical component of developing an optimal recall policy.

A key conclusion of our study is that the optimality of a memory strategy is highly task-dependent. While our work highlights the memory palace as a powerful solution for free recall, its dominance vanished when task demands shifted to require flexible, content-based search. As demonstrated in our conditional free recall experiment, models abandoned the memory palace in favor of a TCM-like strategy when retrieval needed to be guided by item features instead of sequential position. This suggests that the memory palace is specialized for exhaustive sequential retrieval, whereas TCM-like dynamics better support flexible adaptation to diverse, naturalistic demands (Bennion et al., 2025). Future work could test how strategy choice shifts in tasks requiring summaries or contextualization of many items, which may be more useful for guiding future actions than precise episodic detail.

Finally, our model provides a powerful and flexible framework for exploring how episodic memory supports broader cognitive functions. Because the model learns human-like retrieval patterns from basic principles—without hand-crafted cognitive components—it can be readily adapted to study how memory supports higher-order cognition like decision-making, planning, and inference. For instance, future work could extend this model to planning tasks that require retrieving sequences of memories to simulate future outcomes. Such research could bridge our understanding of memory retrieval and its ultimate function in complex, goal-directed behavior.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

4 Methods

4.1 Task

Free recall We used a free-recall task to train the model. There was a study phase and a response phase in each trial. During the study phase, the model was presented with a list of items. During the response phase, the model was trained to recall the studied items in any order. Items were represented as 64-dimensional one-hot vectors, and each trial contained a list of 8 items as the input sequence.

More specifically, during the study phase, the model received one item per time step as input. During the response phase, the model recalled an item per time step as output. A correct output resulted in an immediate reward (+1) to the model. If the model recalled an item that was not presented during the study phase, or if it recalled an item more than once, the outputs were marked as incorrect and the model received a penalty (-0.75). The item list presented during the study phase was generated independently on each trial by randomly sampling 8 items from the full set of 64 items without replacement. There was a time limit of 8 time steps in the response phase, and the model needed to recall all the items correctly to receive the full reward.

Conditional free recall In the conditional free recall, 6 features were assigned to each item and each feature had 2 values. Therefore, there were still 64 items in total. During the study phase, there were 8 items presented as 64-dimensional one-hot vectors for each trial. During the response phase, we gave the model a condition as input. The condition was in the form of "feature x equals to y", in which x took the value of 1 to 6, and y took the value of 1 or 2, both encoded as one-hot vectors. The models needed to recall only items in the list that match this condition. A correct recall led to a reward of +1 and a wrong or repeated recall led to a penalty of -0.75. The output was a 64-dimensional vector representing the probability distribution of recalling each item and an action of recalling an item was sampled from this output, same as the free recall task. The model had 8 time steps to recall for each trial. It could take an action representing "no action" for not outputting any items, and it got no reward or penalty for it. To avoid the model from always taking no action, the model got a penalty of -0.75 for each not-recalled item at the end of the trial if there were items that matched the condition but were not recalled.

The model was trained on this conditional free recall task. During testing, we tested the model on a normal free recall task by setting a feature in all items of the list to be the same and setting the condition to be this feature and its corresponding value.

4.2 Model structure

The model was composed of a 'context module' and a 'memory module'. The 'context module' consisted of 128 gated recurrent units (GRU) (Cho et al., 2014). At each time step, the GRU received inputs and produced outputs with evolving dynamics according to

$$\boldsymbol{h}_t = \phi_\theta(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) \tag{1}$$

$$\boldsymbol{y}_t = f(\boldsymbol{h}_t) \tag{2}$$

Here, ϕ denotes a forward pass through the recurrent GRU dynamics, θ denotes the set of all model parameters, $\boldsymbol{x}_t$ are inputs to the GRU, and $\boldsymbol{y}_t$ are outputs computed from the GRU hidden state $\boldsymbol{h}_t$. The 'memory module' consisted of a list of slots for storing memories. At the start of each trial, the memory module was cleared to allow the storage of new memories, and the hidden state was reset to all zeros.

For each time step in the study phase, the GRU received the one-hot representation of a study item. It then updated the GRU hidden state $\boldsymbol{h}_t$ and appended it to the memory module as an episodic memory $\boldsymbol{m}_t$.

In the response phase, the model retrieved a memory from the memory module and produced an output at every time step. After retrieving a memory, the GRU took the retrieved memory as part of its inputs and updated its hidden state based on the retrieved memory as well as other inputs and the previous hidden

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

state. The output consisted of a 64-dimensional policy $\pi_\theta(a_t)$, which was a set of probabilities associated with each possible item to recall. An action a_t was sampled from the policy as the recalled item.

The memory retrieval process was as follows. At the start of each time step, the model computed the cosine similarity between the current hidden state $\boldsymbol{h}_{t-1}$ and the N memories stored in the memory module $\{\boldsymbol{m}_i | i = 1, ..., N\}$ to produce a similarity vector $\boldsymbol{s}_t$,

$$s_t^i = cos(\boldsymbol{h}_{t-1}, \boldsymbol{m}_i), \qquad i = 1, ..., N. \tag{3}$$

The model then computed the retrieved memory as the weighted sum of all the stored memories according to the weights of the similarity vector. To favor the retrieval of a single memory rather than a combination of multiple memories, we first passed the similarity vector through a softmax function with a low temperature τ,

$$\hat{s}_t^i = \frac{e^{\frac{s_t^i}{\tau}}}{\Sigma_{i=1}^N e^{\frac{s_t^i}{\tau}}}. \tag{4}$$

Then we computed the retrieved memory $\hat{\boldsymbol{m}}_t$ by treating $\hat{\boldsymbol{s}}_t = [\hat{s}_t^1, \hat{s}_t^2, ..., \hat{s}_t^N]$ as weights for each memory,

$$\hat{\boldsymbol{m}}_t = \Sigma_{i=1}^N \hat{s}_t^i \boldsymbol{m}_i. \tag{5}$$

This weighted averaged memory was used instead of the single most similar memory during training to make the memory retrieval process differentiable during training. The softmax temperature of computing memory similarity τ gradually decreases from 0.01 with a rate of 0.9 every 32,000 episodes until reaching less than 0.0002. During testing, we made the model retrieve the single most similar memory instead of a weighted sum to prevent the model from retrieving a uniform combination of all memories. Through gradually reducing the softmax temperature during training, the performance was not affected during testing.

After retrieving a memory, the GRU updated its hidden state on the basis of the retrieved memory as well as the input and previous hidden state.

$$\boldsymbol{z}_t = \sigma(W_z \boldsymbol{x}_t + U_z \boldsymbol{h}_{t-1} + b_z) \tag{6}$$

$$\boldsymbol{r}_t = \sigma(W_r \boldsymbol{x}_t + U_r \boldsymbol{h}_{t-1} + b_r) \tag{7}$$

$$\hat{\boldsymbol{h}}_t = \phi(W_h \boldsymbol{x}_t + U_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \hat{\boldsymbol{m}}_t + b_h) \tag{8}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \hat{\boldsymbol{h}}_t \tag{9}$$

To force the model to use previously stored episodic memories in the memory module instead of working memory stored in the hidden state, the hidden state of the GRU was reset to random noise between the study and response phases as default (proportion of working memory flushed equals to 1). We allowed the model to update its hidden state for a single time step without any recall before retrieving the first memory, so that it could learn an initial state for retrieving the first memory.

4.3 Training algorithm

We trained the model on the free recall task with the advantage actor-critic (A2C) reinforcement learning algorithm (Mnih et al., 2016; Jensen, 2023). The parameters θ updated in the form

$$\Delta\theta \propto \sum_{t=1}^{N} \left[\left(\nabla_\theta \log \pi_t(a_t) + \nabla_\theta V_t \right) \left(-V_t + R_t \right) + \beta_e \nabla_\theta H_a - \beta_m \nabla_\theta H_m \right] \tag{10}$$

Here, $\pi_t(a_t)$ is the policy output by the model at time step t, a_t is the randomly sampled action from the policy (i.e. the item recalled), V_t is the value estimated by the network, and $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_t$, where r_t is the reward at time step t and γ is the discount factor. $H_a = -\sum_a \pi_t(a_t) \log \pi_t(a_t)$ is the entropy of the policy, and β_e is the weight of the entropy regularization term.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

The last reward r_{t-1} and last action a_{t-1} sampled from the policy $\pi_{t-1}(a_{t-1})$ were returned to the model as inputs at the next time step (Wang et al., 2016). This is important for the model to perform the task, since it provides information about what items have been previously recalled, and therefore also which items have yet to be recalled.

To encourage the model to recall only one memory at a time and to inhibit the retrieval of other memories, we added another entropy regularization term $H_m = -\sum_{t=1}^{N}\sum_{i=1}^{N} \hat{s}_t^i \log \hat{s}_t^i$ on the recall weights $\hat{\boldsymbol{s}}_t$ in the loss function. This term encouraged the model to decrease the entropy of the memory similarity $\hat{\boldsymbol{s}}_t$, which made it closer to a one-hot vector. As a consequence, the retrieved memory $\hat{\boldsymbol{m}}_t$ will be closer to a single memory in the memory module instead of a combination of multiple memories.

We trained the neural network model using the Adam optimizer for 1.6×10^6 episodes with a batch size of 16 and a learning rate of 0.003.

4.4 Varying training settings

When we investigated the influence of the proportion of noise injected into working memory between phases, the initial hidden state at the start of the response phase was computed as

$$\boldsymbol{h}_0 = \sqrt{\alpha}\mathcal{N}(0, \sigma^2) + \sqrt{1-\alpha}\boldsymbol{h}'_N \tag{11}$$

Here $\boldsymbol{h}'_N$ is the last hidden state during the study phase. $\mathcal{N}(0, \sigma^2)$ is sampled from a normal distribution in which σ is the standard deviation of $\boldsymbol{h}'_N$. This computation maintains the variance of the hidden state, and α is the proportion of variance replaced by noise in the hidden state, varying from 0 to 1.

We varied the discount factor γ and the noise proportion α from 0 to 1 with an interval of 0.2. For each combination of hyperparameters, we trained 20 models with different random seeds.

When adding noise to the working memory to analyze the noise robustness of the models (Figure 4B,C), we used the same method as above. When noise was added to trained models for testing (Figure 4B), we varied noise proportion from 0 to 0.5 with an interval of 0.02. When noise was added to models during training (Figure 4C), we chose noise proportions of 0.1, 0.04 and 0. The same group of models with different discount factors and noise between phases were trained with different proportion of working memory noise.

4.5 Data analyses

4.5.1 Behavioral metrics for contiguity effect

Conditional recall probability. The conditional recall probability (CRP) curve is a basic method for illustrating the contiguity effect. For an item x_i at position i in the presented list, we computed the probability of recalling the item x_{i+j} immediately after recalling the item x_i, in which j is the lag of item x_{i+j} to the item x_i in the original list. This way we got a probability distribution of recalling the next item as a function of j. We then summed up this distribution for all items by aligning at lag 0, and divided it by a random recall distribution, so that we got the CRP curve.

Forward asymmetry We used the forward asymmetry and the temporal factor to evaluate the contiguity effect of a model. The forward asymmetry (FA) is the tendency to recall in the forward order compared to the backward order. It is defined as the proportion of forward recall transitions in all recall transitions, where a recall transition is defined as a pair of consecutive recall instances. We define $\mathrm{RT}(i, j)$ as a recall transition corresponding to first recalling the ith item and then the jth item in the studied list, and the forward asymmetry can then be computed as

$$\mathrm{FA} = \frac{|\{\mathrm{RT}(i,j)|i<j, i, j = 1, 2, ..., N\}|}{|\{\mathrm{RT}(i,j)|i, j = 1, 2, ..., N\}|} \tag{12}$$

Temporal factor The temporal factor (TF) was introduced by Sederberg et al. (2010) and quantifies the tendency of an agent to recall an item that is in close temporal proximity to the previously recalled item.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

The temporal factor of a recall transition $\mathrm{RT}(i,j)$ in a given sequence of recall behavior is defined as

$$\mathrm{TF}(i,j) = \frac{\mathrm{rank}(\mathrm{RT}(i,j)) - 1}{|\{\mathrm{RT}(i,k)|k \in M_i\}| - 1} \tag{13}$$

Here $M_i = [1,N] \cap \mathbb{Z} \backslash \{m_t | t = 1,2,...,i\}$ is the set of the positions of all items that have not yet been recalled in the list. Therefore, $\{\mathrm{RT}(i,k)|k \in M_i\}$ is the set of all possible correct transitions after recalling the ith item. $\mathrm{rank}(\mathrm{RT}(i,j))$ is the rank of $\mathrm{RT}(i,j)$ in the set $\{\mathrm{RT}(i,k)|k \in M_i\}$, ordered according to the negative absolute value of serial position lag $(-|j-i|)$. When there is a tie, the rank of all members of the tie will be the mean of their ranks. The closer i and j is, the higher $\mathrm{TF}(i,j)$ will be.

4.5.2 Representational analyses of strategy

To quantify the strategy of a model, we used both dimensionality reduction methods and decoding methods. We recorded 5,000 free recall trials from each trained model to perform the analyses.

PCA. We did PCA on the hidden state of the model by concatenating the hidden state in the study phase and the response phase. We fit the PCA on the 5,000 trials and plotted 10 trials for each model.

Decoding index and identity. We used Ridge classifiers to decode the item identity and item index information from the hidden states at each time step. We trained different classifiers for decoding the item identity of items at each time step from GRU hidden states at each time step. For the study phase, items that were studied at a certain time step were combined across trials. For the response phase, items that were recalled at a certain time step were combined across trials.

Explained variance of index and identity. We used linear regression to fit the hidden state with the corresponding index and identity respectively.

$$\boldsymbol{h} = \boldsymbol{\beta}_1 + \boldsymbol{\beta}_{\mathrm{index}} \cdot \boldsymbol{x}_{\mathrm{index}} + \boldsymbol{\epsilon}_1 \tag{14}$$

$$\boldsymbol{h} = \boldsymbol{\beta}_2 + \boldsymbol{\beta}_{\mathrm{identity}} \cdot \boldsymbol{x}_{\mathrm{identity}} + \boldsymbol{\epsilon}_2 \tag{15}$$

in which $\boldsymbol{h}$ is the hidden state, $\boldsymbol{\beta}_{\mathrm{index}}$ and $\boldsymbol{\beta}_{\mathrm{identity}}$ are the weights for index and identity, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are the bias, and $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are the residual error. $\boldsymbol{x}_{\mathrm{index}}$ and $\boldsymbol{x}_{\mathrm{identity}}$ are one-hot representations of index and identity. We then quantified the variance of hidden state explained by index and identity.

$$R^2_{\mathrm{index}} = \frac{SS_{\mathrm{index}}}{SS_{\mathrm{total}}}; R^2_{\mathrm{identity}} = \frac{SS_{\mathrm{identity}}}{SS_{\mathrm{total}}} \tag{16}$$

in which SS_{total} is the total sum of squares of the hidden state samples recorded during testing the model, SS_{index} and SS_{identity} are the regression sum of squares with index and identity. As the index and the identity are independent of each other, the explained variance of the two variables does not overlap. The regression was performed separately for the study phase and response phase, and we averaged the explained variance across the two phases. For the study phase, the identity for each hidden state was the item currently presented, and the index was the presented order of the item. For the response phase, the identity was the currently recalled item, and the index was the index of the recalled item in the original list.

Cross-decoding index and identity For models using the memory palace strategy, hidden representations of index for two different items (from two different trials) with the same index should be similar, and hidden representations of index for the study phase and the response phase should be similar. We measured the level of this item-invariance and phase-invariance of the index code through decoding (Bernardi et al., 2020). For each model, we trained an SVM decoder on hidden states in the study phase with a set of trials including only 8 unique items from the total of 64 items. Then the decoder was tested on hidden states in the response phase with trials generated with all items. We trained another decoder on the response phase and tested on the study phase, and averaged these two decoding results to get the final cross-decoding accuracy of index. We also examined the cross-decoding accuracy of identity. The decoder was trained on hidden states in the study phase with only one same time step from each trial, and was tested on hidden states in the response phase with all time steps of each trial, and vice versa.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

4.5.3 Clustering for strategies

To discover the strategy from the space of the models, we performed a k-means clustering based on the following 9 metrics of each model: forward asymmetry, temporal factor, decoding accuracy of current index, decoding accuracy of current identity, decoding accuracy of the last item identity, explained variance by index and identity, and cross-decoding accuracy of index and identity. Each of the variables was normalized to perform the clustering. Clustering was performed on all models with varying discount factor and proportion of working memory flushed between phases. We filtered out models with free recall performance less than 70%. We tried different numbers of clusters, and computed the inertia and the Sihouette score of each number of clusters (Figure S2A,B). We chose a cluster number of 3 based on the elbow method, which identifies the point where the decrease in inertia began to slow significantly, and referred to the Sihouette score to make sure it became relatively low.

4.6 The key-value memory model

We also trained a model with learnable key/value representations to investigate if our results can be extended to commonly-used key-value memory in machine learning (Zhang et al., 2017; Ritter et al., 2018). It allows the model to use distinct representations for storage content (value) and memory search (keys), which can explain various patterns of the human memory system (Gershman et al., 2025; Song et al., 2025). This distinction might be important for the formation of memory palace strategy in the human brain, where the keys represent the item index, and the values represent the item identity.

During the study phase in the key-value memory model, the hidden state passed a feedforward layer to form a key, and passed another feedforward layer to form a value. Each key-value pair was a memory (Figure S4A). During the response phase, the hidden state passed the same feedforward layer as the key to form a query, and retrieved the value with the highest cosine similarity between the query and the corresponding key. We found that using the same feedforward layer to generate keys and queries is important for contiguity effect and the memory palace strategy to emerge from the model. If the feedforward layer is different for keys and queries, the model cannot show contiguity effect in the first place.

The RNN with a key-value memory model showed the memory palace strategy with the default setting (Figure S4B). We analyzed the explained variance of index and identity respectively on keys and values. We found that the keys represented index more, while the values represented identity more (Figure S4C). PCA trajectories of the hidden state in the study phase and response phase are similar but with some scaling (Figure S4D,E). Keys and values both evolved smoothly with closer states in time more similar (Figure S4F,G). We tried PCA on keys and values together (Figure S4H,I) and separately (Figure S4J,K). While both keys and values represent some index information (Figure S4J,K), they are not in the same space (Figure S4H,I).

4.7 Code availability

Code for this paper is available via GitHub at `https://github.com/Veritaria/rnn-free-recall`.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

References

Anderson, J. R. and Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, 96(4):703–719.

Basile, A. G. and Toplak, M. E. (2015). Four converging measures of temporal discounting and their relationships with intelligence, executive functions, thinking dispositions, and behavioral outcomes. *Frontiers in Psychology*, 6:728.

Bennion, K. A., Venturini, M. C., Reyes, H., Cheng, K., Kendra Eng, T., and Antony, J. W. (2025). Classic free recall memory effects using video stimuli. *Memory*, pages 1–10.

Bernardi, S., Benna, M. K., Rigotti, M., Munuera, J., Fusi, S., and Salzman, C. D. (2020). The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967.

Brown, G. D., Preece, T., and Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological review*, 107(1):127.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dresler, M., Shirer, W. R., Konrad, B. N., Müller, N. C., Wagner, I. C., Fernández, G., Czisch, M., and Greicius, M. D. (2017). Mnemonic training reshapes brain networks to support superior memory. *Neuron*, 93(5):1227–1235.

Gershman, S. J., Fiete, I., and Irie, K. (2025). Key-value memory in the brain. *Neuron*, 113(11):1694–1707.

Glanzer, M. and Cunitz, A. R. (1966). Two storage mechanisms in free recall. *Journal of verbal learning and verbal behavior*, 5(4):351–360.

Glenberg, A. M., Bradley, M. M., Stevenson, J. A., Kraus, T. A., Tkachuk, M. J., Gretz, A. L., Fish, J. H., and Turpin, B. M. (1980). A two-process account of long-term serial position effects. *Journal of Experimental Psychology: Human Learning and Memory*, 6(4):355.

Healey, M. K., Crutchley, P., and Kahana, M. J. (2014). Individual differences in memory search and their relation to intelligence. *Journal of Experimental Psychology: General*, 143(4):1553.

Healey, M. K., Long, N. M., and Kahana, M. J. (2019). Contiguity in episodic memory. *Psychonomic bulletin & review*, 26(3):699–720.

Howard, M. W. and Kahana, M. J. (1999). Contextual variability and serial position effects in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(4):923.

Howard, M. W. and Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of mathematical psychology*, 46(3):269–299.

Jensen, K. T. (2023). An introduction to reinforcement learning for neuroscience. *arXiv preprint arXiv:2311.07315*.

Johnson, J. D. and Rugg, M. D. (2007). Recollection and the reinstatement of encoding-related cortical activity. *Cerebral cortex*, 17(11):2507–2515.

Kahana, M. J. (1996). Associative retrieval processes in free recall. *Memory & cognition*, 24(1):103–109.

Koppenaal, L. and Glanzer, M. (1990). An examination of the continuous distractor task and the "long-term recency effect". *Memory & Cognition*, 18(2):183–195.

Lieder, F. and Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

Logan, G. D. and Cox, G. E. (2021). Serial memory: Putting chains and position codes in context. *Psychological Review*, 128(6):1197.

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

Maguire, E. A., Valentine, E. R., Wilding, J. M., and Kapur, N. (2003). Routes to remembering: the brains behind superior memory. *Nature neuroscience*, 6(1):90–95.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR.

Murdock Jr, B. B. (1962). The serial position effect of free recall. *Journal of experimental psychology*, 64(5):482.

Neath, I. and Crowder, R. G. (1990). Schedules of presentation and temporal distinctiveness in human memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(2):316.

Polyn, S. M. and Kahana, M. J. (2008). Memory search and the neural representation of context. *Trends in cognitive sciences*, 12(1):24–30.

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

Polyn, S. M., Norman, K. A., and Kahana, M. J. (2009b). Task context and organization in free recall. *Neuropsychologia*, 47(11):2158–2163.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770.

Ritter, S., Wang, J., Kurth-Nelson, Z., Jayakumar, S., Blundell, C., Pascanu, R., and Botvinick, M. (2018). Been there, done that: Meta-learning with episodic recall. In *International conference on machine learning*, pages 4354–4363. PMLR.

Roediger, H. L. (1980). The effectiveness of four mnemonics in ordering recall. *Journal of Experimental Psychology: Human Learning and Memory*, 6(5):558.

Sadeghiyeh, H., Wang, S., Alberhasky, M. R., Kyllo, H. M., Shenhav, A., and Wilson, R. C. (2020). Temporal discounting correlates with directed exploration but not with random exploration. *Scientific reports*, 10(1):4020.

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

Sederberg, P. B., Miller, J. F., Howard, M. W., and Kahana, M. J. (2010). The temporal contiguity effect predicts episodic memory performance. *Memory & cognition*, 38:689–699.

Song, H., Lu, Q., Nguyen, T. T., Chen, J., Leong, Y. C., Rosenberg, M. D., Ching, S., and Zacks, J. M. (2025). A neural network with episodic memory learns causal relationships between narrative events. *bioRxiv*, pages 2025–09.

Tan, L., Ward, G., Paulauskaite, L., and Markou, M. (2016). Beginning at the beginning: Recall order and the number of words to be recalled. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(8):1282.

Tulving, E. and Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352.

Unsworth, N. (2019). Individual differences in long-term memory. *Psychological Bulletin*, 145(1):79.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

Wagner, I. C., Konrad, B. N., Schuster, P., Weisig, S., Repantis, D., Ohla, K., Kühn, S., Fernández, G., Steiger, A., Lamm, C., et al. (2021). Durable memories and efficient neural coding through mnemonic training using the method of loci. *Science advances*, 7(10):eabc7606.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

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

Wilding, J. M. and Valentine, E. R. (1997). *Superior memory*. Psychology Press.

Zaromb, F. M., Howard, M. W., Dolan, E. D., Sirotin, Y. B., Tully, M., Wingfield, A., and Kahana, M. J. (2006). Temporal associations and prior-list intrusions in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(4):792.

Zhang, J., Shi, X., King, I., and Yeung, D.-Y. (2017). Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

Zhang, Q., Griffiths, T. L., and Norman, K. A. (2023). Optimal policies for free recall. *Psychological Review*, 130(4):1104.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

5 Extended data

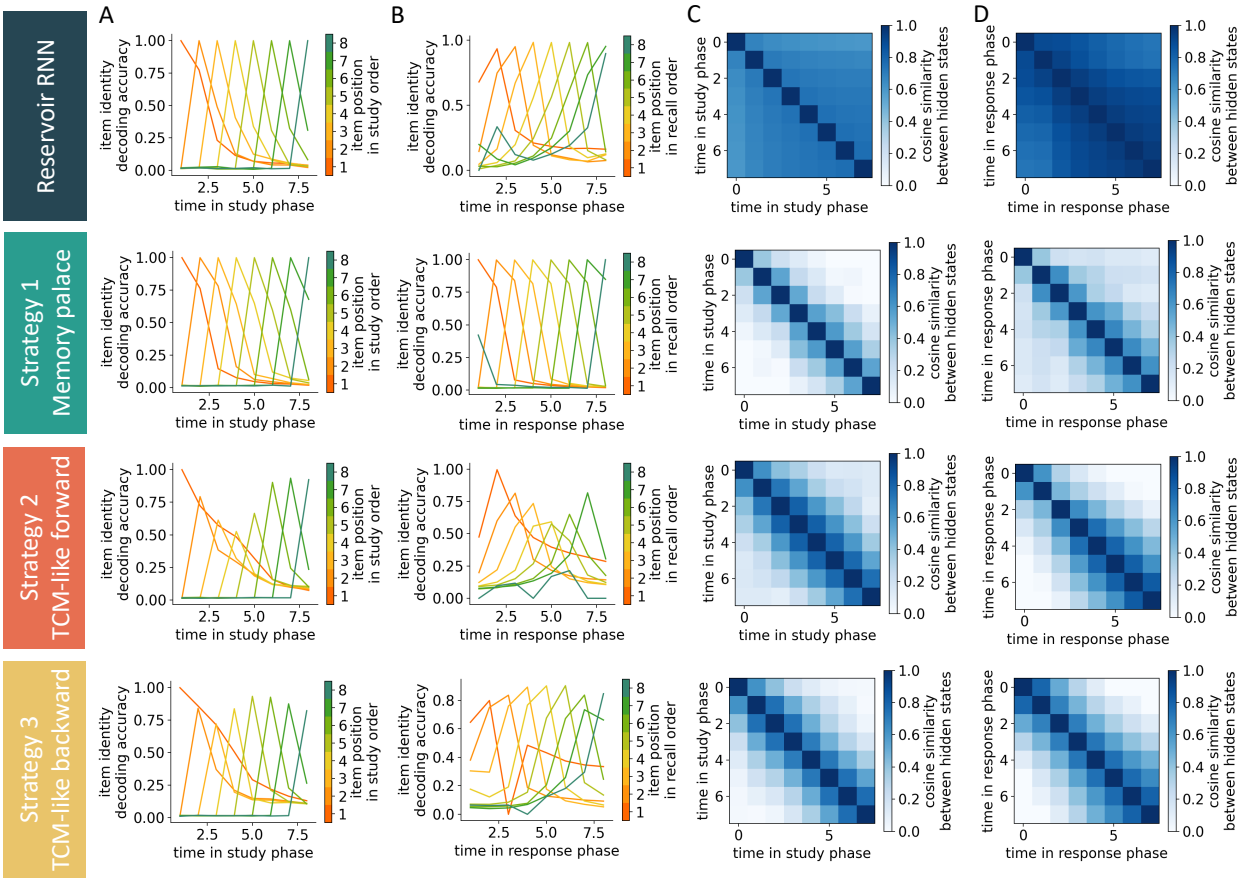


Figure S1: **Supplementary figures for each strategy.** (A) Decoding accuracy of item identity from hidden states during the study phase. Each curve represents items presented at a particular time step. Each data point in the plot is generated with a different decoder, which is trained with cross-validation on data from 1000 trials. (B) Decoding accuracy of item identity from hidden states during the response phase. (C) State similarity between each time step in the study phase. (D) State similarity between each time step in the response phase. For all strategies, the hidden state changes smoothly during both study and response phases.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

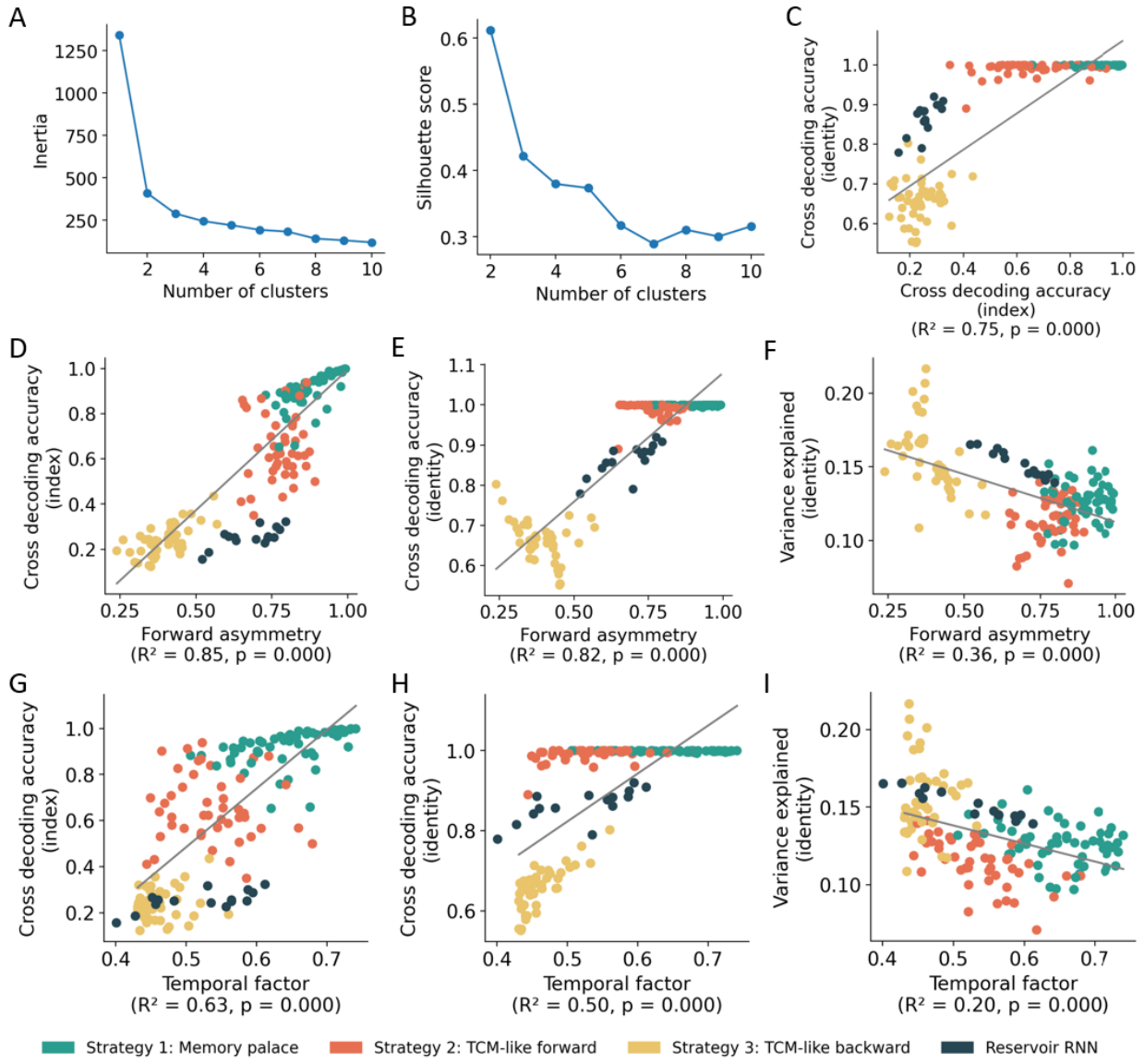


Figure S2: **Supplementary figures for clustering of strategy.** (A) Inertia of clusters in relation with the number of clusters. (B) Sihouette score of different number of clusters. (C) Relation between cross-decoding accuracy of index and identity. (D) Relation between forward asymmetry and cross-decoding accuracy of index. (E) Relation between forward asymmetry and cross-decoding accuracy of identity. (F) Relation between forward asymmetry and explained variance by identity. (G) Relation between temporal factor and cross-decoding accuracy of index. (H) Relation between temporal factor and cross-decoding accuracy of identity. (I) Relation between temporal factor and explained variance by identity.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

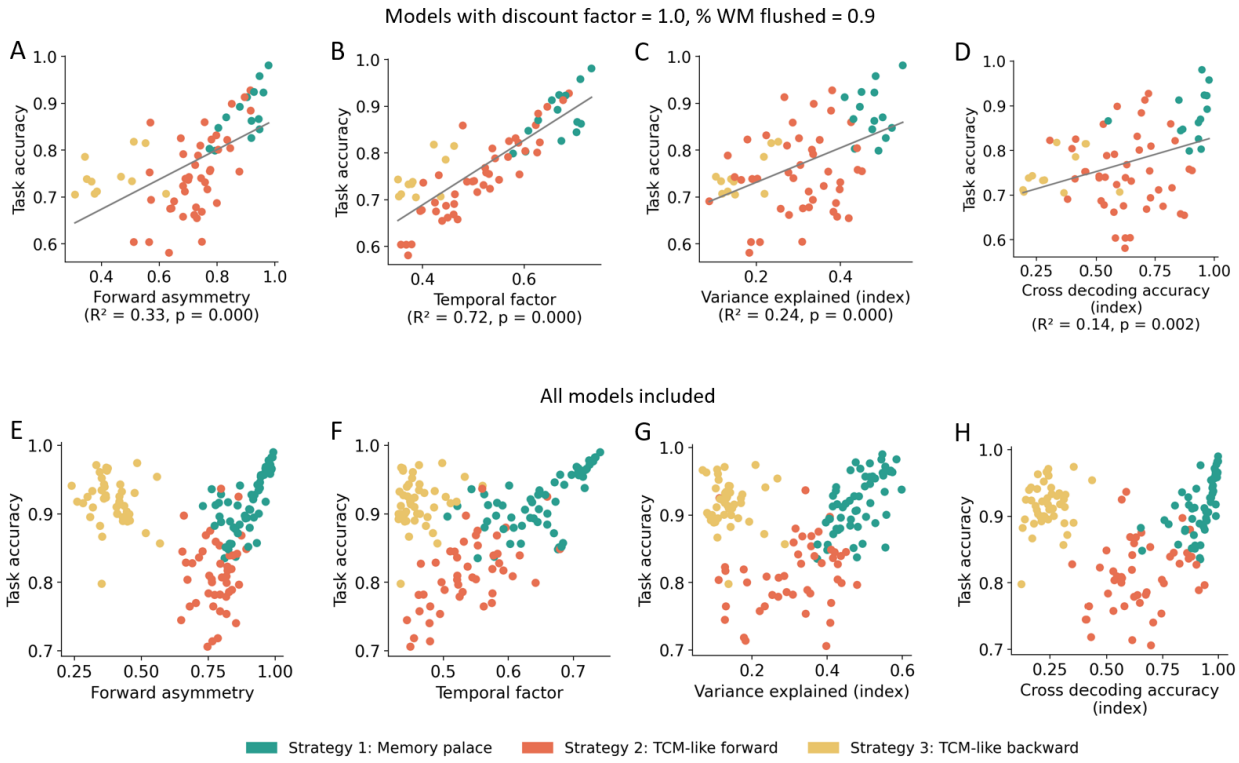


Figure S3: **Relationship between strategy metrics and free recall performance.** Relationship between (A) Forward asymmetry, (B) temporal factor, (C) variance explained by index, (D) cross-decoding accuracy of index and the task performance for 100 models with the same training condition (discount factor = 1.0, % WM flushed = 0.9). Different colors represent different strategies. (E-H) Relationship between the four metrics and the task performance for for all models with different training conditions.

bioRxiv preprint doi: https://doi.org/10.1101/2025.09.25.678592; this version posted September 25, 2025. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY 4.0 International license.

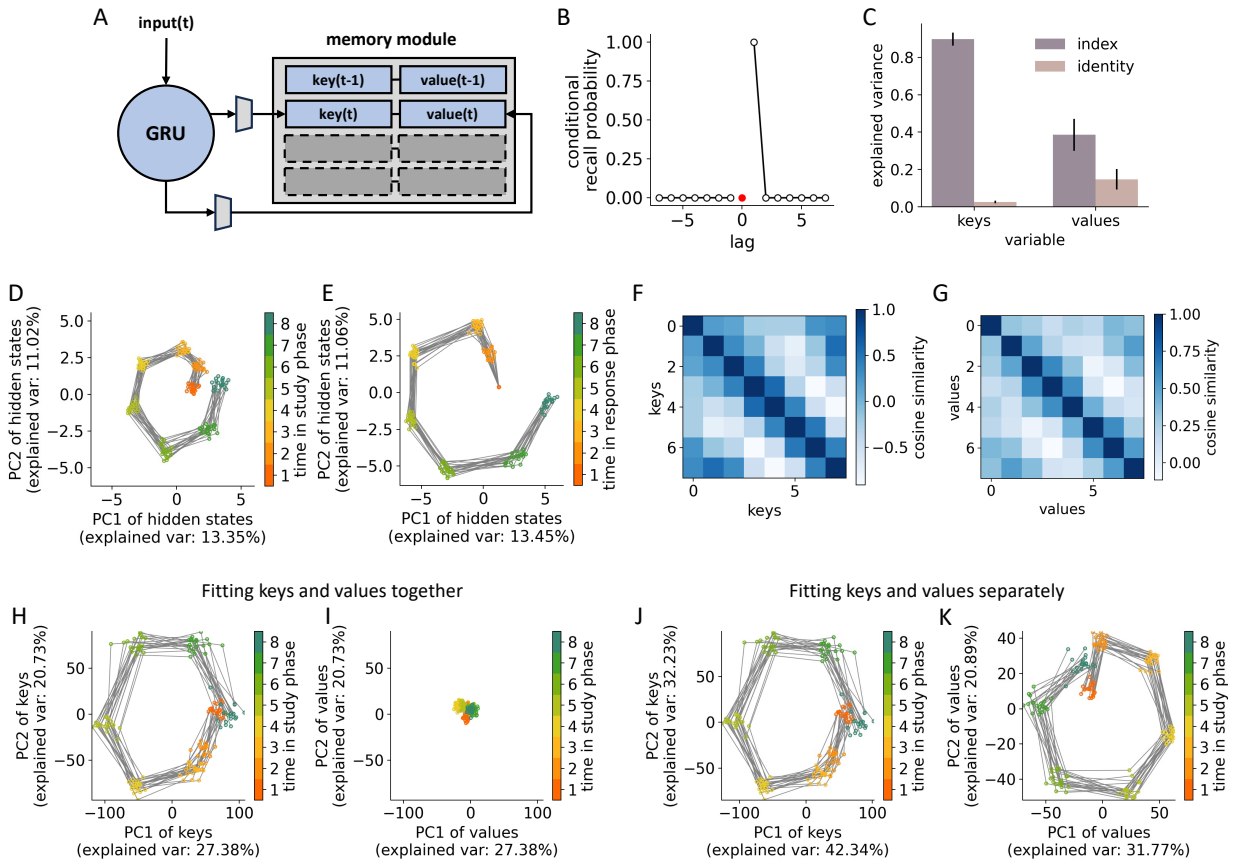


Figure S4: **A key-value memory network showed the memory palace strategy** (A) An illustration of the key-value memory model structure. During the study phase, the GRU hidden state was converted to a key and a value through two different feedforward layers. During the response phase, the GRU passed its hidden state to the same feedforward layer as generating the keys to form a query vector, computed the cosine similarity between the query and all keys, and retrieved the value with the maximum similarity. (B) The conditional recall probability of the key-value memory model with default training settings the same as the previous RNN-memory model. (C) Explained variance of index and identity in keys and values. The keys contained more index information than the values, while the values contained more identity information than the keys. (D) PCA trajectories of the hidden state in the study phase. (E) PCA trajectories of the hidden state in the response phase. (F) Cosine similarity between each key. (G) Cosine similarity between each value. (H, I) PCA trajectories of keys and values when fitting the PCA with them together. (J, K) PCA trajectories of keys and values when fitting them separately.