

Viewpoint

Key-value memory in the brain

Samuel J. Gershman,^{1,2,3,5,*} Ila Fiete,⁴ and Kazuki Irie²¹Department of Psychology, Harvard University, Cambridge, MA, USA²Center for Brain Science, Harvard University, Cambridge, MA, USA³Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University, Cambridge, MA, USA⁴McGovern Institute for Brain Research and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA⁵Lead contact*Correspondence: gershman@fas.harvard.edu<https://doi.org/10.1016/j.neuron.2025.02.029>

SUMMARY

Classical models of memory in psychology and neuroscience rely on similarity-based retrieval of stored patterns, where similarity is a function of retrieval cues and the stored patterns. Although parsimonious, these models do not allow distinct representations for storage and retrieval, despite their distinct computational demands. Key-value memory systems, in contrast, distinguish representations used for storage (values) and those used for retrieval (keys). This allows key-value memory systems to optimize simultaneously for fidelity in storage and discriminability in retrieval. We review the computational foundations of key-value memory, its role in modern machine-learning systems, related ideas from psychology and neuroscience, applications to a number of empirical puzzles, and possible biological implementations.

INTRODUCTION

Despite the apparent fragility of memory, there is no decisive evidence that information, once stored, is ever permanently lost. The storage capacity of the brain must, of course, be finite but that does not seem to be the principal limiting factor on memory performance. Rather, it is the retrieval process that fundamentally limits performance: the relevant information may be there but cannot always be found.^{1–4} Some of the evidence supporting this view will be summarized below.

A retrieval-oriented view of memory performance places the primary explanatory burden on how memories are addressed (i.e., how the retrieval system keeps track of storage locations) and how they are queried (i.e., how the retrieval system maps sensory cues to addresses). There is a long history of theorizing about these concepts in cognitive psychology and neuroscience.^{5,6} Recently, the field of machine learning has developed its own analysis of these concepts, which form the basis of high-performing systems like transformers and fast weight programmers.^{7,8} There is increasing recognition that a significant aspect of intelligence (in both natural and artificial systems) is effective information retrieval.^{9–13}

Our goal is to connect the dots between conceptualizations of memory retrieval in psychology, neuroscience, and machine learning. Central to this effort is the concept of key-value memory, which we formalize below. The basic idea is that inputs (memoranda) are transformed into two distinct representations—keys and values—which are both stored in memory. The keys represent memory addresses, whereas the values store memory content. Memories are accessed by first match-

ing a query to each key and then retrieving a combination of values weighted by their corresponding matches. Importantly, the mappings from inputs into keys and values can be optimized separately, allowing the system to distinguish between information useful for finding memories (stored in the keys) and information useful for answering the query (stored in the values). This distinction is familiar in human-designed information retrieval systems. For example, books often have alphabetically organized indices, which are helpful for finding particular subjects. The indices do not contain any information about the meaning of the subjects themselves; this information is stored in the book's text, retrieved by going to the page number associated with the index.

We will argue that memory in the brain follows similar principles. In particular, we posit a division of labor between a key storage system in the medial temporal lobe and a value storage system in the neocortex. As reviewed below, closely related ideas have already been proposed in psychology and neuroscience. By connecting these ideas to key-value memories, we can begin to understand what makes memory in the brain computationally powerful. To illustrate these points, we present simulations that recapitulate a number of empirical phenomena. Implications for the convergence of natural and artificial intelligence are discussed in the concluding section.

Computational foundations of key-value memory

In this section, we introduce the technical ideas underlying key-value memory, exposing the multitude of ways in which this idea has been conceptualized. We place key-value memory within a modern machine-learning framework by discussing how the

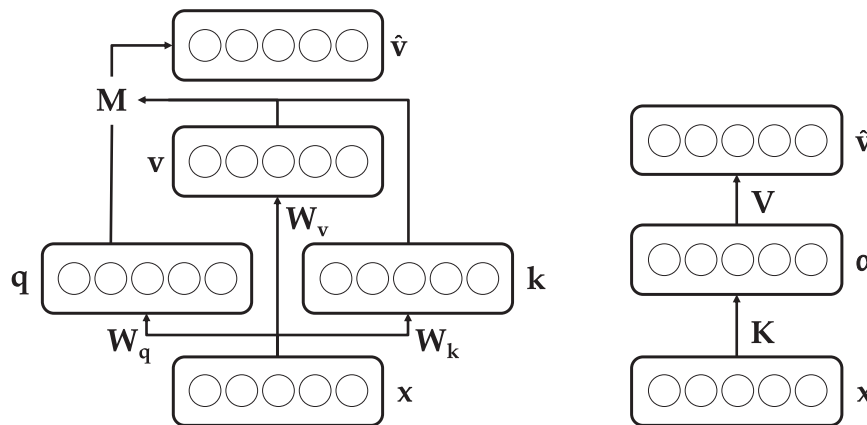


Figure 1. Two architectures for key-value memory

Black symbols denote vectors and blue symbols denote matrices. (Left) Input x is mapped to key (k), query (q), and value (v) vectors. During memory writing, the weight matrix M is updated using Hebbian learning between the key and value vectors. During reading, the query is projected onto M to produce a retrieved value \hat{v} . (Right) The input vector is mapped to a hidden layer α , which is then mapped to an output layer \hat{v} . The input-to-hidden weights correspond to the stored keys; the hidden-to-output weights correspond to the stored values.

key and value representations can be learned, comparing an end-to-end learning approach with fixed (or partially fixed) “scaffolds” for key representations.

From correlations to kernels

One of the earliest formalizations of a key-value memory was Kohonen’s correlation matrix memory model,¹⁴ which was subsequently used by Pike¹⁵ to explain a range of human memory phenomena. Here, we slightly change the notation and terminology to bring this model into correspondence with more recent formalizations (Figure 1, left). Each input, indexed by n , consists of a key vector k_n and a value vector v_n (which we take to be row vectors). Intuitively, the keys encode information about memory addresses (hence we will refer to the set of possible keys as the address space), whereas the values encode information about memory content. These two representations are linked in memory by an “associator” matrix M , which is initialized at 0 and incremented by the outer product of the key and value vectors after each input is presented:

$$\Delta M \propto k_n^T v_n. \quad (\text{Equation 1})$$

It is easy to see that this is just simple Hebbian learning between key and value units encoding the vector elements. In neurobiology, the standard interpretation is that M_{ij} is encoded by the synaptic strength between neurons representing value element j and key element i . The synaptic strength is increased when the two neurons fire coincidentally.¹⁶ Needless to say, learning in the brain is more complicated than this, but, for present purposes, we will take for granted the biological plausibility of Equation 1 (see also Lambacher and Legenstein¹⁷ for a related, but more complex, learning rule).

The correlation matrix memory is heteroassociative because M stores information about the relationship between two different kinds of objects or object properties (see also Steinbuch and Piske¹⁸ for one of the earliest such models). If we impose the constraint that keys and values are the same, we get an autoassociative memory.^{19–22} This idea is now most closely associated with the work of Hopfield,²³ and such models (when the inputs are binary) are typically referred to as Hopfield networks.

The correlation matrix memory is queried by taking the inner product between the associator matrix and a query vector q (with the same dimensionality as the key vectors):

$$\hat{v} = qM. \quad (\text{Equation 2})$$

In neural network terms, this is equivalent to activating pattern q on the key units, which produces an activity pattern in downstream value units based on the learned synaptic strengths (M). The retrieved value vector \hat{v} combines stored values associated with keys that are similar to the query vector. To see this, it is helpful to rewrite Equation 2 in its “dual” form^{24,25}:

$$\hat{v} \propto \sum_{n=1}^N \alpha_n v_n, \quad (\text{Equation 3})$$

where $\{\alpha_n\}$ is a set of “attention weights” computed by $\alpha = \sigma(S(K, q)) = qK^T$ and K is the matrix consisting of all N key vectors (i.e., its rows are k_n). The function $S(\cdot, \cdot)$ is a similarity kernel (linear in this case), expressing the match between keys and queries, and $\sigma(\cdot)$ is a separation operator (the identity function in this case), pushing apart similar memories. The retrieved value vector is thus a weighted combination of stored values, where the attention weights correspond to the similarity between the query and each key, mapped through a separation operator.

The dual form is useful because it shows how the model can be straightforwardly generalized by considering other similarity kernels and separation operators. Any positive semidefinite kernel can be expressed as the inner product in some vector space,²⁶ which means that alternative similarity functions can be constructed by mapping the keys and queries through a feature transform $\phi(\cdot)$ applied to each row:

$$S(K, q) = \phi(q)\phi(K)^T \quad (\text{Equation 4})$$

It is also possible to construct kernels with infinite-dimensional vector spaces by working directly with an explicitly defined similarity kernel, such as the widely used radial basis function kernel, which Tsai and colleagues²⁷ found to produce the best results.

The generalized correlation matrix memory can implement the celebrated self-attention mechanism in transformers⁷:

$$S(\mathbf{K}, \mathbf{q}) = \frac{\mathbf{q}\mathbf{K}^T}{\sqrt{D}} \quad (\text{Equation 5})$$

$$\sigma(\tilde{\alpha}) = \frac{\exp(\tilde{\alpha})}{\sum_n \exp(\tilde{\alpha}_n)} \quad (\text{Equation 6})$$

where D is the dimensionality of the key/value vectors and the softmax function is used as the separation operator. Here, we have used $\tilde{\alpha}$ to denote the attention weights prior to the application of the separation operator. When $\sigma(\cdot)$ is set to the identity function, we recover linearized attention,²⁵ which can be expressed as a special form of recurrent neural network known as a fast weight programmer.^{8,28} Linearized attention has a potential computational advantage over softmax attention because its recurrent form allows for linear-complexity sequence processing, while giving up the parallel computation property of the attention form (whose complexity, however, is quadratic in the number of sequence elements). Several studies^{29,30} have shown how to rigorously approximate softmax attention using random features.

As shown by Millidge and colleagues,³¹ a variety of classical memory models can also be derived by different choices of similarity kernel and separation function. For example, sparse distributed memory³² can be obtained by setting $\sigma(\cdot)$ to a threshold function, and dense associative memory³³ can be obtained by setting $\sigma(\cdot)$ to a rectified polynomial.

In the noiseless setting, the ideal separation function is the max operator because this will always return the stored value associated with the matching key.³¹ However, this separation function is not robust to noise—small perturbations can cause large retrieval errors. Thus, the design of a memory system needs to balance separability and robustness. In addition, the memory may be used for generalization to new inputs,^{7,34,35} in which case perfect matching is not the goal. For example, key-value memories have been used extensively for question-answering tasks,^{36,37} where \mathbf{q} represents a question and \mathbf{v} represents an answer. The system stores a set of question-answer pairs and tries to use this knowledge base to answer novel questions.

Representational structure

So far, our setup has assumed a memory system provided with keys, values, and queries—where do these representations come from? In modern machine learning, they are derived as mappings from input vectors, $\{\mathbf{x}_n\}$. A typical assumption in transformers and fast weight programming is to parametrize the mappings with a set of linear functions:

$$\mathbf{k}_n = \mathbf{x}_n \mathbf{W}_k \quad (\text{Equation 7})$$

$$\mathbf{v}_n = \mathbf{x}_n \mathbf{W}_v \quad (\text{Equation 8})$$

$$\mathbf{q}_n = \mathbf{x}_n \mathbf{W}_q \quad (\text{Equation 9})$$

where $\{\mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_q\}$ are learned weight matrices. The input vectors may themselves be learned embeddings of raw input data,

harnessing the power of deep learning systems trained end-to-end.

In some systems, the key and query mappings are assumed to be fixed, either randomly or based on some regular structure. In these systems, the address space is conceived as a kind of “scaffold” for the indexing of information content. In the sparse distributed memory model,³² the scaffold is random so that similar keys do not systematically index similar values. This mimics (approximately) the organization of random-access memory in digital computers. A familiar example of a structured scaffold is the alphabetical index found at the end of books. In the Hopfield network²³ and related autoassociative memory models, the scaffold is identical to the value space. This property endows the models with content addressability³⁸: memory addresses are accessed by matching values directly to queries. Most psychological and neural models of memory share this property, though they implement it in different ways.

Some models assume that the scaffold is randomly constructed in such a way that it carries a structural imprint. For example, if inputs are assigned to random addresses, but these addresses change slowly over time, then inputs experienced nearby in time will be encoded as more “similar” in the address space. Temporal autocorrelation, in the absence of additional structure, is able to account for many aspects of human³⁹ and animal⁴⁰ memory.

Below, we will discuss neurobiologically inspired architectures that implement a fixed scaffold, which can even outperform a learned mapping. We will also show that learning the key and query mappings allows us to explain data on repulsion effects in memory and hippocampal representations. These results pose a number of questions for future research: does the brain use a combination of fixed and learned mappings? Is such a combination useful for machine-learning applications?

The ubiquity of key-value memory

The work reviewed above focused on explicit constructions of key-value memory. It turns out that other models are sometimes implicitly equivalent. Irie and colleagues¹² showed that linear layers trained by gradient descent (ubiquitous in many machine-learning models) can also be expressed as key-value memories. Here, we briefly review this theorem.

Let \mathbf{x}_n be the input vector at time n , as in the previous section. The output of a linear layer is $\mathbf{y}_n = \mathbf{x}_n \mathbf{W}$, where \mathbf{W} is a weight matrix. The weight matrix is trained by gradient descent, yielding the following after N timesteps:

$$\mathbf{W} = \mathbf{W}_0 + \sum_{n=1}^N \mathbf{x}_n^T \mathbf{e}_n, \quad (\text{Equation 10})$$

where \mathbf{W}_0 is the initial weight matrix and $\mathbf{e}_n = -\eta_n (\nabla_{\mathbf{y}} \mathcal{L})_n$ is the error signal with learning rate η_n and loss function \mathcal{L} . Generalizing a classic result on kernel machines,⁴¹ Irie and colleagues¹² showed that this construction is equivalent (in the sense of producing the same output for a given input) to the following linear key-value memory:

$$\mathbf{y} = \mathbf{x} \mathbf{W}_0 + \sum_{n=1}^N \alpha_n \mathbf{v}_n, \quad (\text{Equation 11})$$

where (using our earlier notation) $\alpha = \mathbf{q}\mathbf{K}^T$ with $\mathbf{v}_n = \mathbf{e}_n$, $\mathbf{k}_n = \mathbf{x}_n$, and $\mathbf{q} = \mathbf{x}$, given an arbitrary input \mathbf{x} . Thus, linear layers effectively memorize their experienced error patterns, computing their outputs as linear functions of these memories. This interpretation is intriguing in light of the fact that errors are particularly salient in human memory.^{42–46}

The linear layers retain information about all training inputs—they never “forget.”¹² However, retrieval access may be (transiently) lost. We discuss evidence for this idea from psychology and neuroscience below.

Neurobiological substrates

Although key-value memories are loosely inspired by the brain, it remains an open question how to implement them in a biologically plausible manner. Equation 1 is a (somewhat) biologically plausible rule for learning associations between keys and values, but we still need rules for storing the keys and values themselves. In the section on representation learning, we described the widely used approach of modeling keys, queries, and values as linear mappings from input vectors (which themselves could be learned). This implies additional learning rules for the linear mappings. One could posit that they are learned by backpropagating the errors from whatever objective function is being optimized for a downstream task.¹⁷ This would require a biologically plausible approximation of the backpropagation algorithm.⁴⁷

Kozachkov and colleagues⁴⁸ have proposed an architecture based on the “tripartite synapse,” consisting of pre-synaptic and post-synaptic neurons modulated by an astrocyte (a type of glial cell). In particular, they posited that this motif applies to the hidden-to-output synapses in the three-layer network implementation of key-value memory. The activation of each astrocyte process is modeled as a linear function of the hidden unit activations. They showed how the astrocyte processes collectively compute the similarity function $S(\mathbf{K}, \mathbf{q})$, which then multiplicatively modulate the hidden-to-output weights so that the network as a whole implements the transformer self-attention described earlier.

Alternatively, Tyulmankov and colleagues⁴⁹ have proposed a non-Hebbian learning rule for key learning. They view the key-value memory as a three-layer neural network, where the input \mathbf{x} (first layer) is transformed into a pattern of attention α (hidden layer), which is finally transformed into retrieved values $\hat{\mathbf{v}}$ (output layer); see the right panel of Figure 1. Each hidden-layer unit represents a “slot” to which a single input (or possibly multiple inputs) gets assigned. In this view, the key matrix \mathbf{K} corresponds to the input-hidden synaptic strengths and the value matrix \mathbf{V} (where row n corresponds to value vector \mathbf{v}_n) corresponds to the hidden-output synaptic strengths. The proposed learning rule for the synapse connecting input unit j to hidden unit i is (simplifying slightly and dropping time indices):

$$\Delta K_{ij} \propto \mu \gamma_i (x_j - K_{ij}), \quad (\text{Equation 12})$$

where $\mu \in \{0, 1\}$ is a global third factor (possibly a neuromodulator like acetylcholine or dopamine) and $\gamma_i \in \{0, 1\}$ is a local third factor (possibly a dendritic spike). The factors are binary to promote sparsity in the hidden layer. The local third factor tags the least-recently used hidden unit as eligible for plasticity. The au-

thors point out that the key learning rule resembles behavioral timescale plasticity,⁵⁰ which has been observed in the hippocampus. The possibility that key learning occurs in the hippocampus will be considered in detail below.

For value learning, Tzyulmankov and colleagues clamp the output layer to the target value \mathbf{v} and then update the synapse connecting hidden unit i to output unit m according to the following:

$$\Delta V_{mi} \propto \mu \gamma_i \alpha_i (v_m - V_{mi}). \quad (\text{Equation 13})$$

This is a Hebbian rule because it depends on the co-activation of hidden and output units. It could potentially describe modification of the output projections from the hippocampal area CA1 to the entorhinal cortex.

Using a similar architecture (but with the important addition of recurrence), Whittington and colleagues⁵¹ have shown that the Tolman-Eichenbaum Machine,⁵² a model of the entorhinal-hippocampal system, implements a form of key-value memory. The hippocampus, according to this model, stores conjunctive representations of sensory inputs from the lateral entorhinal cortex and a “structural code” provided by cells in the medial entorhinal cortex, which are updated recurrently. Conjunctive representations in the hippocampus function as both the keys and the values (i.e., an autoassociative memory). In a two-dimensional (2D) open field, the model can reproduce hippocampal “place cells” (firing fields localized to a particular spatial location), while the structural code generates grid-cell-like responses (grid cells fire with hexagonal periodicity as an animal moves through space⁵³). In environments of different dimensions, topologies, and geometries, the structural code need not be grid-like. Whittington and colleagues note that the structural code in their model functions as a form of “position encoding”—widely used in transformers and other sequence models—where the encodings (i.e., addresses) are adapted to the structure of space. This is a critical aspect of the model: the structure-sensitive properties of their modeled hippocampal and entorhinal cells would not have been obtained using the fixed positional encodings typically used in regular transformers. Related ideas have proven successful in the machine learning literature.⁵⁴

In contrast to the adaptive addresses used in the Tolman-Eichenbaum Machine, Sharma, Chandra, and colleagues^{55,56} proposed neurobiologically motivated models (MESH and Vector-HaSH) that use random projections of modular fixed point (attractor) networks to address memories (see Box 1 for details). The fixed points within each modular attractor, which are fixed and not learned, may be sparse and random⁵⁶ or possess a 2D geometry⁵⁵ such as the hexagonally periodic activity patterns of grid cells. Consistent with the non-learned assumption, grid cells exhibit a fixed relational organization regardless of environment and behavioral state.^{57–60} The same relationships are re-used to encode the organization of non-spatial variables.^{61–63} The fixed modular attractor networks in the models randomly and densely project into a larger network (the hippocampus in Vector-HaSH) with self-consistent return projections, forming a large scaffold of quasi-random stable fixed point address states. Thus, we may view MESH and Vector-HaSH as fitting the diagram of Figure 1 (right), except

Box 1. Random projections of attractor states for error-correcting key-query matching

Vector-HaSH⁵⁵ and MESH⁵⁶ are tripartite networks consisting of an input (sensory) layer, a densely connected layer, and a layer consisting of a modular set of fixed recurrent attractor networks. These models can be directly connected with a key-query-value formulation: define the tripartite network weights to be \mathbf{W}_{sd} , \mathbf{W}_{da} , \mathbf{W}_{ad} , \mathbf{W}_{ds} (from the sensory layer s to a densely connected layer d , dense to attractor layer a , attractor to dense, and dense to sensory, respectively). Define the attractor, dense layer, and attractor states to be $\{\mathbf{a}\}$, $\{\mathbf{d}\}$, $\{\mathbf{s}\}$, respectively. If we view the dense layer responses as linear (in practice they are nonlinear), the network performs the following operation:

$\hat{\mathbf{s}} = \phi(\mathbf{s}\mathbf{W}_{sd}\mathbf{W}_{da}, \{\mathbf{a}\})\mathbf{W}_{ad}\mathbf{W}_{ds}$ where $\phi(\cdot, \cdot)$ is an efficient nearest-neighbor (NN) operation performed on the sensory inputs by the scaffold consisting of the recurrent attractor network and the dense layer. The attractor retrieves the nearest of the attractor states $\{\mathbf{a}\}$ to the input $\mathbf{s}\mathbf{W}_{sd}\mathbf{W}_{da}$. This model can be interpreted as the key-value system of Figure 1B, in which the output of the attractor layer can be defined as a (denoised) query $\mathbf{q}_n = \mathbf{x}_n = \phi(\mathbf{s}\mathbf{W}_{sd}\mathbf{W}_{da}, \{\mathbf{a}\})$, and \mathbf{W}_{ad} and \mathbf{W}_{ds} correspond to the key and value matrices, respectively (i.e., $\mathbf{K} = \mathbf{W}_{ad}^T$ and $\mathbf{V} = \mathbf{W}_{ds}$); that is, the diagram's hidden states (attention weights) are $\alpha_n = \mathbf{x}_n\mathbf{W}_{ad}$. The combined attractor-dense layer scaffold computes a similarity $\sigma(\mathbf{S}(\mathbf{K}, \mathbf{q}))$ and the network produces the output $\hat{\mathbf{v}}_n = \hat{\mathbf{s}}_n = \alpha_n\mathbf{W}_{ds} = \mathbf{x}_n\mathbf{W}_{ad}\mathbf{W}_{ds} = \phi(\mathbf{s}_n\mathbf{W}_{sd}\mathbf{W}_{da}, \{\mathbf{a}\})\mathbf{W}_{ad}\mathbf{W}_{ds} = \sigma(\mathbf{S}(\mathbf{K}, \mathbf{q}_n))\mathbf{W}_{ds}$.

The value weights (from the dense to the sensory layer) and part of the query weights (from the sensory to the dense layer) are trained by a Hebbian rule so that $\mathbf{W}_{sd} = \sum_n \mathbf{s}_n^T \mathbf{d}_n$ and $\mathbf{W}_{ds} = \sum_n \mathbf{d}_n^T \mathbf{s}_n$; the weights \mathbf{W}_{da} are random and fixed and \mathbf{W}_{ad} are also held fixed (after being set as a pseudoinverse on the random \mathbf{W}_{da} , so these may also be viewed as random). Thus, the key weights are random and fixed; the choice of modular attractor states and random projections from attractor neurons to the dense layer pre-defines keys as a well-separated set so that NN behaves well for noisy inputs. The query weights are partly random, fixed, and coupled with the keys and partly learned through Hebbian plasticity.

that the hidden layer in the diagram is replaced by a bipartite scaffold circuit (modular attractor network bidirectionally coupled to the densely connected layer) that computes nonlinear robust nearest-neighbor (NN) search to compute $\sigma(\mathbf{S}(\mathbf{K}, \mathbf{Q}))$, with keys that are random and fixed. In Vector-HaSH, keys reside in hippocampus and the memory values are reconstructed at the sensory input layer (cortex).

The advantages of this construction, beyond standard key-value constructions, are that (1) the address space is large and (2) the scaffold fixed points possess large and uniform basins of attraction. These properties allow many memories to be robustly addressed, with error correction and without collapse (blurring together the addresses of multiple memories). Such collapse is characteristic of the “memory cliff” that afflicts many memory models^{64,65}: after a threshold number of inputs have been stored, retrieval performance crashes. In contrast, MESH and Vector-HaSH produce graceful degradation as multiple memories share overlapping addresses, much like human memory.⁶⁶ Remarkably, these models also outperform a flexible encoder trained to minimize reconstruction error,⁶⁷ suggesting that the entorhinal-hippocampal system may be a highly effective memory addressing system discovered by evolution.

The 2D and geometric aspect of Vector-HaSH over MESH becomes relevant for memorizing (possibly discrete) items embedded, remembered, or retrieved through traversal of a continuous low-dimensional space. This may involve sequential episodic memories (where time is the continuous dimension) or spatial memories (where space is the continuous dimension).

Evidence from psychology and neuroscience

In this section, we review several lines of evidence suggestive of key-value memory in the brain. Our review is organized around the following claims central to the theory. (1) Memories are stored indelibly but subject to retrieval interference. (2) Memories are addressed by representations (keys) that are distinct from

the representations of memory content (values). The representational structure of the keys is optimized for discriminability, whereas the representational structure of the values is optimized for fidelity. (3) The information stored in keys is not available to conscious access (i.e., recall). In other words, the brain uses keys to recall values but cannot recall the keys themselves.

Retrieval interference, not erasure, is the principal limiting factor in memory performance

An important implication of key-value memory systems is that memory performance is constrained primarily by the ability to retrieve relevant information, not by storage capacity. In this section, we review some of the theoretical arguments and empirical evidence that this assumption can be plausibly applied to the brain.

A number of attempts have been made to estimate the storage capacity of the human brain.⁶⁸ Depending on different assumptions about numbers of neurons and connectivity, estimates can range from 10^7 to 10^{15} bits. The total number of bits arriving from sensory systems with minimal compression are estimated to range from 10^{13} to 10^{17} . These rough numbers suggest that, with adequate compression, storage capacity may not be the strongest constraint on memory performance.

Perhaps more persuasively than these theoretical arguments, we can make the case based on observations about behavior. If memory storage has reached its capacity limit, then it is impossible to store new information without removing some old information. This implies that old information should become permanently inaccessible at some point. In contrast, studies of human memory demonstrate that memories can be stored over decades, despite being rarely rehearsed.^{69–72}

One might try to explain these findings by arguing that memory storage has not reached the capacity limit, but then it would be very challenging to explain forgetting over the much shorter intervals studied in experiments on list memory. When presented with a list of random words, the proportion of recalled words

declines with list length (typically in the range of 5 to 20 items). If this was due to removal of items from memory, then one would expect catastrophic forgetting over intervals of decades.

Furthermore, a series of experiments by Shiffrin⁷³ demonstrated that, when presented with a sequence of lists and then asked to recall the list before the most recently presented list, performance depended not on the length of the most recent list but only on the length of the list being recalled. This suggests that forgetting is not due to displacement of old items by items from the last list. The limiting factor is retrieval interference from other items on the same list.

Using word-location pairs, Berens and colleagues⁷⁴ separately estimated memory accessibility (whether or not a location is recalled at all given a word cue) and precision (the variance of angular errors). They found that accessibility, but not precision, declined as a function of the retention interval. Similar results have been reported using memory for real-world events.⁷⁵ Thus, memories do not melt into oblivion but rather disappear from view. When they come back into view, they are as sharp as they were before they disappeared.

Supposedly lost memories can be found when appropriate retrieval cues are used,⁷⁶ when the number of retrieval attempts increases,^{77,78} or even spontaneously after a sufficiently long delay.⁷⁹ This holds not only for standard memory tasks in healthy subjects but also for retrograde amnesia induced experimentally or by neurological damage. Spontaneous “shrinkage” of amnesia is a common clinical observation following brain trauma,⁸⁰ presumably due to the restoration of memory access. In laboratory studies, amnesia can be induced experimentally in a range of ways, such as electroconvulsive shock, hypothermia, protein synthesis inhibition, and lesion or inactivation of the hippocampus, with recovery also induced in a range of ways.^{4,81,82} For example, protein synthesis inhibition following classical conditioning typically eliminates conditioned responding on long-term memory tests. However, delayed testing sometimes reveals recovery of performance.^{83–85} It is even possible to restore performance using the amnesic agent itself.^{86–88}

These observations about amnesia mirror well-known phenomena in classical conditioning. Extinction of a previously conditioned stimulus (i.e., presenting the stimulus in the absence of the unconditioned stimulus) causes a decline in conditioned responding, eventually reaching baseline levels. This decline is transient: conditioned responding can return spontaneously⁸⁹ or can be induced by a single “reminder” of the unconditioned stimulus.⁹⁰

In summary, considerable evidence suggests that failures of remembering primarily arise from failures of retrieval, typically due to interference from other memories. Memories thought to be lost can later be found under the right conditions. We will explore this idea computationally when we discuss model simulations.

Distinct representations of keys and values

The influential “Complementary Learning Systems” framework holds that there is a division of labor between the hippocampus and neocortex,^{91,92} with the hippocampus specialized for episodic memory (remembering events that occurred in a specific spatiotemporal context) and a set of neocortical areas (sometimes referred to as “association cortex”) that are special-

ized for semantic memory (remembering regularities that generalize across episodes). This framework has also influenced the design of artificial intelligence systems.⁹³ In this section, we will argue that this framework can be understood in terms of key-value memory.

Rather than thinking about the hippocampus as directly storing memory content, we can alternatively conceptualize it as storing keys and matching queries to keys that address memory content stored in neocortex. This view emphasizes the point that episodic memories need to be bound to semantic content—otherwise, they are essentially empty vessels, as in cases of semantic dementia, where degeneration of the anterior temporal lobe and prefrontal areas produces profound semantic impairments despite relatively intact recognition memory for recent experiences.⁹⁴

If the hippocampus provides the keys necessary for activating neocortical values, then we would expect a causal interplay between the two. Indeed, there is empirical evidence for the following facts: (1) cortical-encoding-related activity is reinstated at the time of memory retrieval, (2) cortical reinstatement depends on the hippocampus, and (3) the reinstatement is necessary for memory retrieval.^{95–100}

Another line of evidence comes from studies of generalization. In the absence of the hippocampus, neocortical values cannot be accessed in a targeted way, leading to overgeneralization. A study by Winocur and colleagues¹⁰¹ offers a good example. When trained to anticipate a shock in context A, rats specifically freeze in context A but not in context B when tested a day later. When tested a week later, rats show a generalization effect (loss of context specificity), freezing in both contexts. This generalization might arise because of interference from memories acquired during the intervening time. Context specificity can be restored by “reminding” the rat of the original memory (briefly placing it back in context A). This reminder effect can be interpreted as activation of the appropriate address given a highly specific cue. Importantly, the reminder effect disappears in hippocampal-lesioned rats, consistent with the claim that the hippocampus stores the keys necessary for targeting specific memories (see also Wiltgen et al.¹⁰² for converging evidence).

The idea that the hippocampus stores keys was to a large extent anticipated by the hippocampal memory indexing theory,¹⁰³ which proposed that the hippocampus indexes memory content stored in neocortical areas. Since it was first proposed, new techniques have uncovered much more detailed support for the theory.^{104,105} In particular, the advent of activity-dependent labeling and optogenetic manipulation has enabled the identification of “engram cells” in the hippocampus that are causally linked to specific memories.^{106,107} Goode and colleagues¹⁰⁵ interpret hippocampal engrams as indices (in the sense of Teyler and DiScenna), linking together information stored in a distributed network of neocortical areas.

Several brain-wide engram mapping studies in rodents have reported a collection of neocortical (as well as subcortical) areas that are active during both encoding and retrieval and therefore qualify as engrams.^{108,109} What makes the hippocampus special is its role as a hub region with high connectivity to neocortical regions.¹¹⁰ This allows the hippocampus to exert strong control over neocortical regions. In addition, hippocampal engrams

are highly sparse (involving a small subset of hippocampal cells) and conjunctively tuned to multiple sensory and cognitive inputs; these features make hippocampal engrams well-suited to encoding episodically unique memories. A recent study of food-caching birds¹¹¹ provides a particularly dramatic demonstration: hippocampal ensembles encoded unique representations for over 100 cache sites (including multiple caches at the same location), which were reactivated upon memory retrieval.

If hippocampal representations are optimized for discriminating between distinct episodes in a cue-dependent manner, then we should expect changes in these representations under different retrieval demands. For example, overlapping routes in spatial navigation need to be discriminated in order to avoid confusion. Chanales and colleagues¹¹² showed that this situation produces repulsion of hippocampal representations specifically for overlapping locations (see also Wanjia et al.¹¹³). The repulsion effect emerges gradually over the course of learning, ultimately reversing the objective similarity relations between locations. The strength of the repulsion effect also correlated with behaviorally measured discrimination accuracy.

Note that optimization for discriminability is only part of the story because the hippocampus receives noisy inputs that may activate the wrong keys. The hippocampus needs to do some error correction/pattern completion in order to “clean up” the retrieved keys. This may be accomplished by attractor dynamics such that nearby inputs get mapped to a corresponding attractor (pattern completion), whereas more distant inputs are separated into distinct attractors (pattern separation). It remains to be seen whether such attractor dynamics are implemented by the recurrent hippocampal-entorhinal loop⁵⁵ (Box 1) or by the proposal that both occur within the hippocampus, with pattern separation by dentate gyrus and pattern completion by area CA3.¹¹⁴

In summary, evidence suggests a division of labor between key encoding in the hippocampus and value encoding in the neocortex. Hippocampal keys are optimized for discrimination, whereas neocortical values are optimized for encoding semantic regularities.

Values, but not keys, are available for recall

Keys store information that is never overtly recalled. Testing this hypothesis is challenging because it is possible that some information stored in keys is also stored in values. Clear support comes from evidence that there is information used to guide retrieval (putatively stored in keys) that, nonetheless, is not available for overt recall. More technically, we characterize “overt recall” of a key as the ability to report some aspect of the key vector. The evidence presented below suggests that keys can be matched to queries without making the content of the keys available for report.

Many of us are familiar with the experience of having a memory at the “tip of the tongue”—stored in memory yet temporarily unrecallable.^{115,116} Closely related is the “feeling of knowing”—a subjective judgment about the likelihood of subsequently recognizing items that are presently unrecallable. The typical study procedure is to present subjects with difficult general knowledge questions, elicit feelings of knowing for questions they cannot presently answer, and then subsequently test their ability to recognize correct answers to the questions. An important finding from this literature is that feelings of knowing predict (though not

perfectly) subsequent recognition,^{117,118} indicating that people are able to judge whether some piece of information is stored in memory despite not being able to retrieve it. Similar results have been found with cued recall tests.¹¹⁹ Another clue comes from studies examining response times: Reder¹²⁰ found that people could report answerability for trivia questions faster than they could report the answers, again indicating that meta-memory does not require retrieval of memory content. Furthermore, feelings of knowing (but not recall) can be spuriously increased by increasing cue familiarity,^{121,122} possibly due to increased key-query match without increased probability that the correct keys are matched.

These phenomena are broadly consistent with the hypothesis that key-query matching can be used to judge whether some information (e.g., the answer to a question) is stored in memory, without accessing the memory content (value) itself. This idea has appeared in the human long-term-memory literature under various guises; for example, Koriat¹²³ discuss the tip-of-the-tongue phenomenon in terms of “pointers” (cues that specify a memory address without specifying memory content), whereas Morton and colleagues¹²⁴ use “headings” to refer to essentially the same idea.

A pointer system has also been invoked to understand short-term memory: rather than storing transient copies of long-term memories, short-term memory might store pointers that refer to information in long-term memory.^{125,126} This may explain why people can detect changes even when they cannot report what exactly has changed,¹²⁷ analogous to the “butcher on the bus” phenomenon in long term memory,¹²⁸ where people can recognize familiar items without being able to recollect any details about them. Both change detection without identification and familiarity without recollection might be accomplished using keys (pointers) that are content addressable but do not obligatorily activate their associated values.⁵⁵

Many standard models of memory cannot check whether information is stored without retrieving that information at least partially. This is because most models base recognition memory judgments on the match between the cue and stored content; there is no separate representation of keys that can be used to check whether something is stored in memory. This makes it challenging for these models to explain why people can be knowledgeable about what is stored in memory without recalling it.

RESULTS

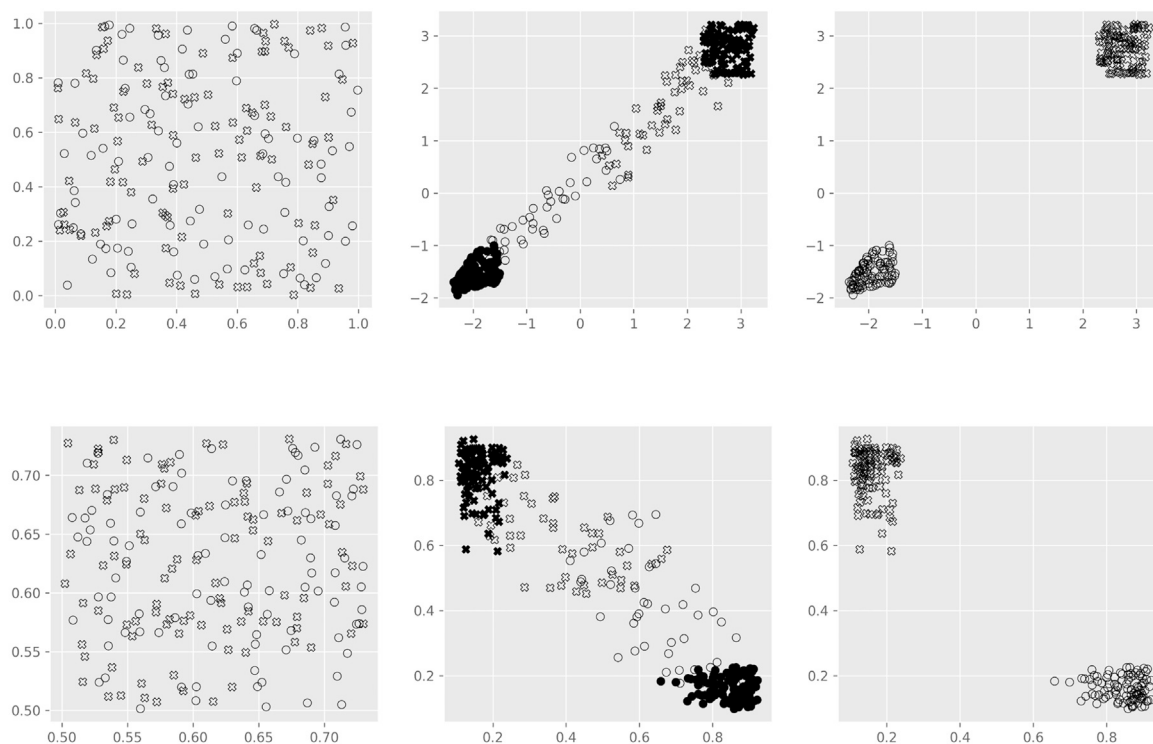
Illustrative simulations

In this section, we provide two simulations that illustrate some of the distinctive characteristics of key-value models highlighted above. Further implementation details can be found in our public code repository, available at <https://github.com/kazuki-irie/kv-memory-brain>.

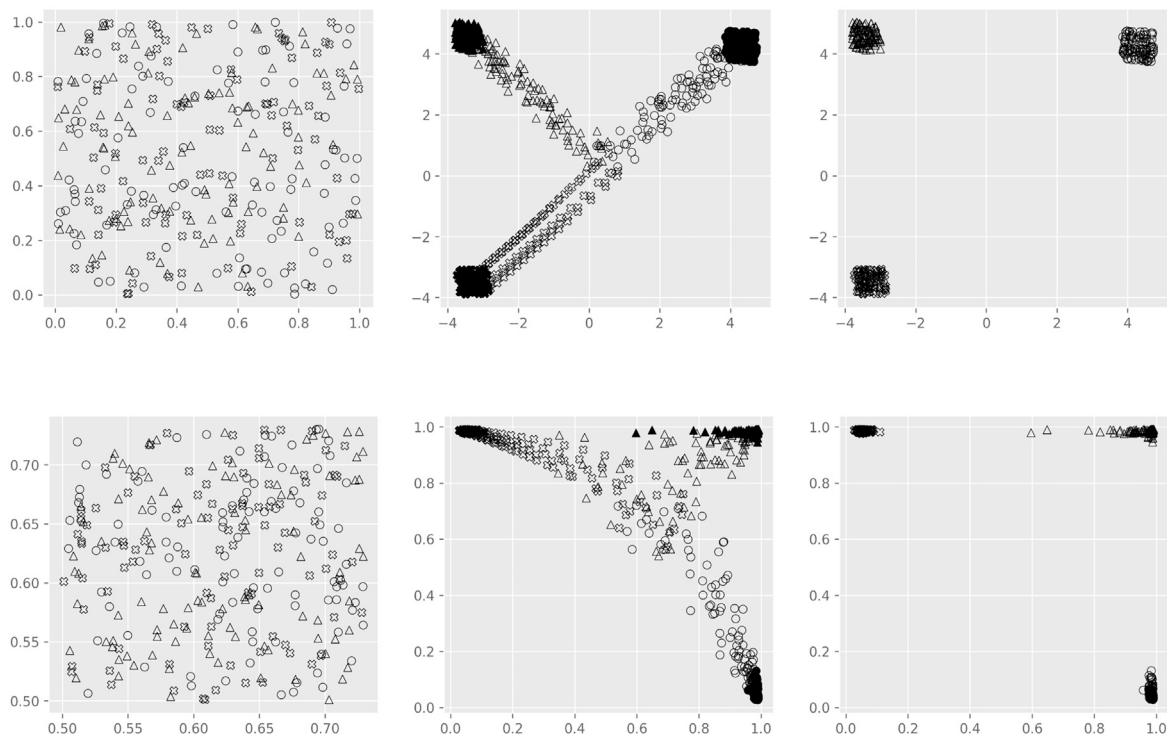
Distinct representations for keys and values

As we discussed earlier, one of the essential properties of key-value memory is the separate representations allocated for keys and values, which can be optimized for their specific roles in retrieval and storage, respectively. Here, we present a toy simulation that illustrates this property.

A



B



(legend on next page)

We consider a minimal key-value model whose trainable parameters are a set of pairs of key and value vectors (i.e., in this model, we skip the step of mapping inputs to keys and values and directly examine the properties of key/value representations). We set both keys and values to be 2D vectors that can be easily visualized in the 2D space. The model can take an arbitrary 2D vector as an input that is treated as a query; the query is compared with all the keys through dot product to obtain a similarity score for each key (as in Equation 5). The resulting scores are normalized by applying the softmax function (Equation 6) to obtain the final “attention” scores. The output of the model is the weighted average of value vectors using these attention scores (Equation 2).

To train the model, we randomly assign each key/value pair to a class; we test two settings using either two or three classes in total. Each class has a fixed feature vector (representing some specific object or event): in the two-class case, the feature vectors for class “0” and “1” are vectors (0, 1) and (1, 0) in the 2D space, respectively; in the three-class case, we additionally have a third class, class “2” with (1, 1) as its feature vector. The task of the model is to output the correct class feature vector when one of the keys is fed to the model as input. We apply the sigmoid function to the value vectors to ensure their effective values are between 0 and 1. The key and value vectors are initialized with a uniform distribution between 0 and 1, and our goal is to examine what key and value representations emerge when this key-value model is optimized for this simple retrieval task. We use the mean squared error loss and the gradient descent algorithm, as is commonly used in modern deep learning.

The results are shown in Figure 2. We observe that key and value representations effectively exhibit different optimal configurations. The key configuration is optimized for softmax-based discrimination, facilitating effective retrieval: as we can see in Figure 2A, top row (the two-class case), the two classes take two opposite quadrants (which are highly distinguishable through dot product and softmax). The trend is similar for the three-class case (Figure 2B, top row). In contrast, the value representations are optimized to represent the class features (i.e., the memory content). For example, in the two-class case (Figure 2A, bottom row), they are optimized to represent (0, 1) for class “blue” and (1, 0) for class “red.” Again, the trend is similar for the three-class case (Figure 2B, bottom row). This illustrates how the key-value memory architecture allows for separate representations for keys and values, optimized for retrieval and storage, respectively.

Forgetting as retrieval failure, and recovery by memory reactivation

Another distinctive property of the key-value memory we highlighted earlier is that, as recall relies on successful retrieval, forgetting can be conceptualized as retrieval failure; that is, even when a memory trace (i.e., a key/value pair corresponding to an event) is stored in memory, forgetting can still occur when

retrieval fails due to interference/inaccessibility. This also implies that, if we manage to fix the failure in the retrieval process, there is a hope for recovering the corresponding memory without having to relive the event itself. Here, we present a simulation illustrating these phenomena using an artificial neural network that learns two tasks sequentially (the so-called continual learning scenario) and examine it through the lens of key-value memory. We also highlight how this simple experiment echoes neurobiological findings on optogenetic recovery of memory following retrograde amnesia.^{129,130}

We train a simple feedforward neural network on two binary image classification tasks sequentially. Using two classic image datasets commonly used in deep learning, MNIST, and FashionMNIST,^{131,132} we construct the two toy tasks that involve classifying images from the two first classes of MNIST (digit 0 vs. 1) and FashionMNIST (“T-shirt” vs. “Trouser”), respectively. The model has one hidden layer (i.e., it has two linear transformations: input-to-hidden and hidden-to-output mappings, with a rectifier linear activation function in-between). The input images are grayscale and their dimensions are 28×28 ; they are flattened to yield a 784-dimensional vector accepted by the input layer. We set the hidden-layer dimension to 64 and the model output size is 4 (for two times two-way classification tasks). The model is trained on the cross-entropy loss using the gradient descent algorithm; by applying the dual formulation described earlier, each linear layer in the model can be formalized as a key-value system keeping memory traces of the entire learning experience by storing layer inputs as keys and error signals as values for every learning step.

In this sequential learning setting, we first train the model on task 1 (MNIST) for 5 epochs, after which the model is trained for another 5 epochs on the training dataset of task 2 (FashionMNIST) without having access to the task 1 training data anymore. Although such a training process produces a set of weight matrices for neural networks in their conventional form, in the key-value memory view, the final model consists of a sequence of key/value vectors; in this specific scenario of two-task sequential learning, each key/value pair belongs to either task 1 (MNIST) or task 2 (FashionMNIST) learning experiences.

Figure 3A shows the evolution of model performance on the test set of the two tasks as a function of training epochs. We observe that the model achieves ~99% test accuracy on task 1 (MNIST) in the first 5 epochs corresponding to task 1 learning, but this performance drops to ~9% after learning task 2 (FashionMNIST), until the model achieves ~95% test accuracy on task 2 (we deliberately train the model long enough to observe forgetting). This amnesic phenomenon, reminiscent of catastrophic forgetting in neural networks,^{65,133,134} is intriguing given the key-value memory view of the model in which the key/value memory pairs belonging to task 1 remain part of the model parameters, explicitly and indefinitely. By using the same

Figure 2. Optimization of key and value representations

Each point represents an event in the memory and belongs to one of (A) two or (B) three classes, represented by different shapes. In each case, the evolution of key (top row) and value (bottom row) representations during the optimization process is shown; each row shows (left) random initialization and (middle) trajectory of representations during the optimization process; (right) final configuration. We observe that keys are optimized for retrieval/separability, whereas values are optimized to store the memory content.

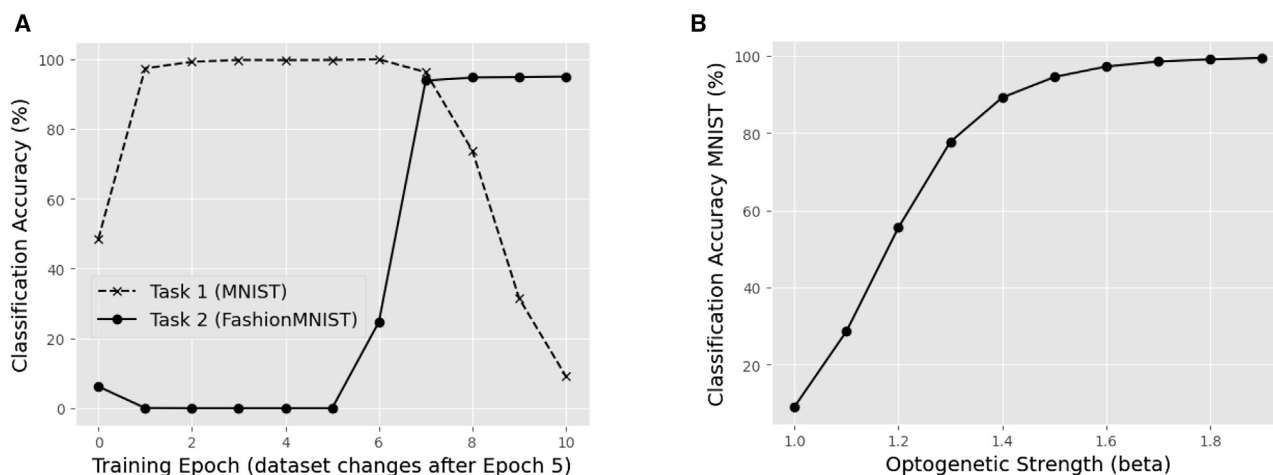


Figure 3. Forgetting and reactivation of memory events

A one-layer feedforward neural network is trained on two tasks sequentially, tasks 1 and 2, constructed using the MNIST and FashionMNIST datasets, respectively. (A) The evolution of the test classification accuracy for the two tasks as a function of training epochs. After epoch 5, the training dataset changes from task 1 to task 2, resulting in forgetting of task 1 as the model learns task 2. (B) The accuracy of the trained model on task 1 as a function of the value of the artificial scalar β used to amplify the keys in all key-value memory pairs corresponding to task 1 learning.

terminology as in neuroscience experiments,¹²⁹ these task 1-related key/value memories became “silent” after task 2 learning.

An interesting question is whether we can reactivate these silent key/value memories to recover the model’s performance on task 1 without any retraining on task 1, akin to how experimental neuroscientists successfully reactivated “silent engrams” through an optogenetic procedure.¹³⁰ For this, we introduce a single positive scalar $\beta \geq 1$ (an “optogenetic strength”) to multiply/amplify the keys (or, equivalently in this model, the values) in all key-value pairs corresponding to task 1 in the trained model. Figure 3B shows the performance of the model on task 1 (MNIST) as a function of the optogenetic strength β ($\beta = 1$ corresponds to no intervention on the model). We observe that, by simply increasing β (i.e., by artificially “reactivating” the existing key-value memories corresponding to the model’s prior MNIST learning experiences), the model becomes capable of solving MNIST again without any retraining.

This simulation not only illustrates the core property of the key-value memory, where forgetting can be attributed to retrieval failure, but also resonates with the neuroscientific findings of silent memory engrams in retrograde amnesia and their recovery through artificial reactivation.

Conclusions

Our goal in this paper was to connect ideas about key-value memory across artificial intelligence, cognitive science, and neuroscience. We have argued that the brain might plausibly implement a key-value memory system in the division of labor between hippocampus and neocortex. We have also argued that a number of behavioral findings from memory studies (e.g., the ability to report item familiarity without recollection) is consistent with a key-value architecture.

Presently, the connections we have highlighted are speculative. On the empirical side, we hope that more direct experi-

mental tests will be undertaken. For example, the key-value architecture implies that repulsion effects in long-term memory should be reversible—a prediction that has not yet been tested. On the theoretical side, we hope that more biologically detailed models will be developed that can explain the plethora of findings discussed in previous sections.

Our paper was motivated by the fact that key-value memory seems to be an important ingredient in the success of several modern artificial intelligence systems, such as transformers and fast weight programmers. The idea that the brain may implement something like this indicates evidence for convergence¹³⁵ and suggests that this is a promising direction for exploring brain-like mechanisms that can power intelligent systems.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to the lead contact, Samuel J. Gershman (gershman@fas.harvard.edu).

Materials availability

This study did not generate new materials.

Data and code availability

Our code is publicly available online at <https://github.com/kazuki-irie/kv-memory-brain>. This repository contains the code required to reproduce all the simulation results presented in this paper, including Figures 2 and 3.

ACKNOWLEDGMENTS

The authors are grateful for support from the Kempner Institute for the Study of Natural and Artificial Intelligence and from the Department of Defense MURI program under ARO grant W911NF-23-1-0277.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
 - Distinct representations for keys and values
 - Forgetting as retrieval failure, and recovery by memory reactivation

Received: January 6, 2025

Revised: February 14, 2025

Accepted: February 28, 2025

Published: March 26, 2025

REFERENCES

1. Tulving, E. (1974). Cue-dependent forgetting. *Am. Sci.* 62, 74–82.
2. Crowder, R.G. (1976). *Principles of Learning and Memory* (Lawrence Erlbaum).
3. Lewis, D.J. (1979). Psychobiology of active and inactive memory. *Psychol. Bull.* 86, 1054–1083. <https://doi.org/10.1037/0033-2909.86.5.1054>.
4. Miller, R.R. (2021). Failures of memory and the fate of forgotten memories. *Neurobiol. Learn. Mem.* 181, 107426. <https://doi.org/10.1016/j.nlm.2021.107426>.
5. Kahana, M.J. (2012). *Foundations of Human Memory* (Oxford University Press).
6. Chaudhuri, R., and Fiete, I. (2016). Computational principles of memory. *Nat. Neurosci.* 19, 394–403. <https://doi.org/10.1038/nn.4237>.
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. *Adv. Neural Inf. Process. Syst.* 30.
8. Schmidhuber, J. (1992). Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Comput.* 4, 131–139. <https://doi.org/10.1162/neco.1992.4.1.131>.
9. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 471–476. <https://doi.org/10.1038/nature20101>.
10. Gershman, S.J., and Daw, N.D. (2017). Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annu. Rev. Psychol.* 68, 101–128. <https://doi.org/10.1146/annurev-psych-122414-033625>.
11. Geva, M., Schuster, R., Berant, J., and Levy, O. (2021). Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495.
12. Irie, K., Csordás, R., and Schmidhuber, J. (2022). The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *International Conference on Machine Learning*, pp. 9639–9659.
13. Allen-Zhu, Z., and Li, Y. (2024). Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. In *Forty-first International Conference on Machine Learning* <https://openreview.net/forum?id=5x788rqbcj>.
14. Kohonen, T. (1972). Correlation matrix memories. *IEEE Trans. Comput.* C-21, 353–359. <https://doi.org/10.1109/TC.1972.5008975>.
15. Pike, R. (1984). Comparison of convolution and matrix distributed memory systems for associative recall and recognition. *Psychol. Rev.* 91, 281–294. <https://doi.org/10.1037/0033-295X.91.3.281>.
16. Caporale, N., and Dan, Y. (2008). Spike Timing-Dependent Plasticity: A Hebbian Learning Rule. *Annu. Rev. Neurosci.* 31, 25–46. <https://doi.org/10.1146/annurev.neuro.31.060407.125639>.
17. Limbacher, T., and Legenstein, R. (2020). H-mem: Harnessing synaptic plasticity with hebbian memory networks. *Adv. Neural Inf. Process. Syst.* 33, 21627–21637.
18. Steinbuch, K., and Piske, U.A.W. (1963). Learning Matrices and Their Applications. *IEEE Trans. Electron. Comput.* 6, 846–862. <https://doi.org/10.1109/PGEC.1963.263588>.
19. Willshaw, D.J., Buneman, O.P., and Longuet-Higgins, H.C. (1969). Non-holographic associative memory. *Nature* 222, 960–962. <https://doi.org/10.1038/222960a0>.
20. Anderson, J.A. (1970). Two models for memory organization using interacting traces. *Math. Biosci.* 8, 137–160. [https://doi.org/10.1016/0025-5564\(70\)90147-1](https://doi.org/10.1016/0025-5564(70)90147-1).
21. Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Trans. Comput.* C-21, 1197–1206. <https://doi.org/10.1109/T-C.1972.223477>.
22. Nakano, K. (1972). Associatron—a model of associative memory. *IEEE Trans. Syst. Man Cybern. SMC-2*, 380–388. <https://doi.org/10.1109/TSMC.1972.4309133>.
23. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79, 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>.
24. Ba, J., Hinton, G.E., Mnih, V., Leibo, J.Z., and Ionescu, C. (2016). Using fast weights to attend to the recent past in. *Adv. Neural Inf. Process. Syst.* 29, 4331–4339.
25. Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the 37th International Conference on Machine Learning*, 119, pp. 5156–5165. <https://proceedings.mlr.press/v119/katharopoulos20a.html>.
26. Schölkopf, B. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press).
27. Tsai, Y.-H.H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. (2019). Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics), pp. 4344–4353. <https://aclanthology.org/D19-1443>.
28. Schlag, I., Irie, K., and Schmidhuber, J. (2021). Linear Transformers Are Secretly Fast Weight Programmers. In *Proceedings of the 38th International Conference on Machine Learning*, 139, pp. 9355–9366. <https://proceedings.mlr.press/v139/schlag21a.html>.
29. Choromanski, K.M., Likhoshesterov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, Q., Mohiuddin, A., Kaiser, L., et al. (2021). Rethinking Attention with Performers. In *International Conference on Learning Representations* <https://openreview.net/forum?id=Ua6zuk0WRH>.
30. Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, A., and Kong, L. (2021). Random Feature Attention. In *International Conference on Learning Representations*. <https://doi.org/10.48550/arXiv.2103.02143>.
31. Millidge, B., Salvatori, T., Song, Y., Lukaszewicz, T., and Bogacz, R. (2022). Universal Hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pp. 15561–15583.
32. Kanerva, P. (1988). *Sparse Distributed Memory* (MIT Press).
33. Krotov, D., and Hopfield, J.J. (2016). Dense Associative Memory for Pattern Recognition in *Advances in Neural J. Inf. Process. Syst.* 29. https://proceedings.neurips.cc/paper_files/paper/2016/file/eaee339c4d89fc102edd9dbdb6a28915-Paper.pdf.

34. Ramsauer, H., Schafl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D., Kopp, K., et al. (2021). Hopfield Networks is All You Need. In International Conference on Learning Representations <https://openreview.net/forum?id=tL89RnzliCd>.
35. Bricken, T., Davies, X., Singh, D., Krotov, D., and Kreiman, G. (2023). Sparse Distributed Memory is a Continual Learner. In International Conference on Learning Representations <https://openreview.net/forum?id=JknGeelZJpHP>.
36. Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-To-End Memory Networks. *Adv. Neural J. Inf. Process. Syst.* 28. https://proceedings.neurips.cc/paper_files/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.
37. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., and Weston, J. (2016). Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1400–1409.
38. Kohonen, T. (1980). Content-Addressable Memories (Springer Science & Business Media). <https://doi.org/10.1007/978-3-642-96552-4>.
39. Landauer, T.K. (1975). Memory without organization: Properties of a model with random storage and undirected retrieval. *Cogn. Psychol.* 7, 495–531. [https://doi.org/10.1016/0010-0285\(75\)90020-1](https://doi.org/10.1016/0010-0285(75)90020-1).
40. Estes, W.K. (1955). Statistical theory of spontaneous recovery and regression. *Psychol. Rev.* 62, 145–154. <https://doi.org/10.1037/h0048509>.
41. Aizerman, M., Braverman, E., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Autom. Remote Control* 25, 821–837.
42. Green, R.T. (1956). Surprise as a factor in the von Restorff effect. *J. Exp. Psychol.* 52, 340–344. <https://doi.org/10.1037/h0047496>.
43. Hirshman, E., Whelley, M.M., and Palij, M. (1989). An investigation of paradoxical memory effects. *J. Mem. Lang.* 28, 594–609. [https://doi.org/10.1016/0749-596X\(89\)90015-6](https://doi.org/10.1016/0749-596X(89)90015-6).
44. Sakamoto, Y., and Love, B.C. (2004). Schematic Influences on Category Learning and Recognition Memory. *J. Exp. Psychol. Gen.* 133, 534–553. <https://doi.org/10.1037/0096-3445.133.4.534>.
45. Rouhani, N., Norman, K.A., and Niv, Y. (2018). Dissociable effects of surprising rewards on learning and memory. *J. Exp. Psychol. Learn. Mem. Cogn.* 44, 1430–1443. <https://doi.org/10.1037/xlm0000518>.
46. Bein, O., Plotkin, N.A., and Davachi, L. (2021). Mnemonic prediction errors promote detailed memories. *Learn. Mem.* 28, 422–434. <https://doi.org/10.1101/lm.053410.121>.
47. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., and Hinton, G. (2020). Backpropagation and the brain. *Nat. Rev. Neurosci.* 21, 335–346. <https://doi.org/10.1038/s41583-020-0277-3>.
48. Kozachkov, L., Kastanenko, K.V., and Krotov, D. (2023). Building transformers from neurons and astrocytes. *Proc. Natl. Acad. Sci. USA* 120, e2219150120. <https://doi.org/10.1073/pnas.2219150120>.
49. Tyulmankov, D., Fang, C., Vadaparty, A., and Yang, G.R. (2021). Biological learning in key-value memory networks. *Adv. Neural Inf. Process. Syst.* 34, 22247–22258.
50. Bittner, K.C., Milstein, A.D., Grienberger, C., Romani, S., and Magee, J.C. (2017). Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* 357, 1033–1036. <https://doi.org/10.1126/science.aan3846>.
51. Whittington, J.C.R., Warren, J., and Behrens, T.E. (2022). Relating transformers to models and neural representations of the hippocampal formation. In International Conference on Learning Representations <https://openreview.net/forum?id=B8DV09B1YE0>.
52. Whittington, J.C.R., Muller, T.H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T.E.J. (2020). The Tolman-Eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell* 183, 1249–1263.e23. <https://doi.org/10.1016/j.cell.2020.10.024>.
53. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E.I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature* 436, 801–806. <https://doi.org/10.1038/nature03721>.
54. Liu, X., Yu, H.-F., Dhillon, I., and Hsieh, C.-J. (2020). Learning to encode position for transformer with continuous dynamical model. In *International Conference on Machine Learning*, pp. 6327–6335.
55. Chandra, S., Sharma, S., Chaudhuri, R., and Fiete, I. (2025). Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature* 638, 739–751. <https://doi.org/10.1038/s41586-024-08392-y>.
56. Sharma, S., Chandra, S., and Fiete, I. (2022). Content addressable memory without catastrophic forgetting by heteroassociation with a fixed scaffold. In *International Conference on Machine Learning*, pp. 19658–19682.
57. Yoon, K., Buice, M.A., Barry, C., Hayman, R., Burgess, N., and Fiete, I.R. (2013). Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat. Neurosci.* 16, 1077–1084. <https://doi.org/10.1038/nn.3450>.
58. Gardner, R.J., Lu, L., Wernle, T., Moser, M.-B., and Moser, E.I. (2019). Correlation structure of grid cells is preserved during sleep. *Nat. Neurosci.* 22, 598–608. <https://doi.org/10.1038/s41593-019-0360-0>.
59. Trettel, S.G., Trimper, J.B., Hwaun, E., Fiete, I.R., and Colgin, L.L. (2019). Grid cell co-activity patterns during sleep reflect spatial overlap of grid fields during active behaviors. *Nat. Neurosci.* 22, 609–617. <https://doi.org/10.1038/s41593-019-0359-6>.
60. Gardner, R.J., Hermansen, E., Pachitariu, M., Burak, Y., Baas, N.A., Dunn, B.A., Moser, M.B., and Moser, E.I. (2022). Toroidal topology of population activity in grid cells. *Nature* 602, 123–128. <https://doi.org/10.1038/s41586-021-04268-7>.
61. Constantinescu, A.O., O'Reilly, J.X., and Behrens, T.E.J. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science* 352, 1464–1468. <https://doi.org/10.1126/science.aaf0941>.
62. Aronov, D., Nevers, R., and Tank, D.W. (2017). Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit. *Nature* 543, 719–722. <https://doi.org/10.1038/nature21692>.
63. Killian, N.J., Jutras, M.J., and Buffalo, E.A. (2012). A map of visual space in the primate entorhinal cortex. *Nature* 491, 761–764. <https://doi.org/10.1038/nature11587>.
64. Amit, D.J., Gutfreund, H., and Sompolinsky, H. (1987). Statistical mechanics of neural networks near saturation. *Ann. Phys.* 173, 30–67. [https://doi.org/10.1016/0003-4916\(87\)90092-3](https://doi.org/10.1016/0003-4916(87)90092-3).
65. McCloskey, M., and Cohen, N.J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychol. Learn. Motiv.* 24, 109–165.
66. Barnes, J.M., and Underwood, B.J. (1959). Fate of first-list associations in transfer theory. *J. Exp. Psychol.* 58, 97–105. <https://doi.org/10.1037/h0047507>.
67. Radhakrishnan, A., Belkin, M., and Uhler, C. (2020). Overparameterized neural networks implement associative memory. *Proc. Natl. Acad. Sci. USA* 117, 27162–27170. <https://doi.org/10.1073/pnas.2005013117>.
68. Dudai, Y. (1997). How big is human memory, or on being just useful enough. *Learn. Mem.* 3, 341–365. <https://doi.org/10.1101/lm.3.5.341>.
69. Bahrnick, H.P., Bahrnick, P.O., and Wittlinger, R.P. (1975). Fifty years of memory for names and faces: A cross-sectional approach. *J. Exp. Psychol.: Gen.* 104, 54–75. <https://doi.org/10.1037/0096-3445.104.1.54>.
70. Bahrnick, H.P., and Hall, L.K. (1991). Lifetime maintenance of high school mathematics content. *J. Exp. Psychol.: Gen.* 120, 20–33. <https://doi.org/10.1037/0096-3445.120.1.20>.
71. Conway, M.A., Cohen, G., and Stanhope, N. (1991). On the very long-term retention of knowledge acquired through formal education: Twelve years of cognitive psychology. *J. Exp. Psychol.: Gen.* 120, 395–409. <https://doi.org/10.1037/0096-3445.120.4.395>.

72. Maxcey, A.M., Shiffrin, R.M., Cousineau, D., and Atkinson, R.C. (2022). Two case studies of very long-term retention. *Psychon. Bull. Rev.* 29, 563–567. <https://doi.org/10.3758/s13423-021-02002-y>.
73. Shiffrin, R.M. (1970). Forgetting: Trace erosion or retrieval failure? *Science* 168, 1601–1603. <https://doi.org/10.1126/science.168.3939.1601>.
74. Berens, S.C., Richards, B.A., and Horner, A.J. (2020). Dissociating memory accessibility and precision in forgetting. *Nat. Hum. Behav.* 4, 866–877. <https://doi.org/10.1038/s41562-020-0888-8>.
75. Diamond, N.B., Armon, M.J., and Levine, B. (2020). The truth is out there: accuracy in recall of verifiable real-world events. *Psychol. Sci.* 31, 1544–1556. <https://doi.org/10.1177/0956797620954812>.
76. Wagenaar, W.A. (1986). My memory: A study of autobiographical memory over six years. *Cogn. Psychol.* 18, 225–252. [https://doi.org/10.1016/0010-0285\(86\)90013-7](https://doi.org/10.1016/0010-0285(86)90013-7).
77. Buschke, H. (1974). Spontaneous remembering after recall failure. *Science* 184, 579–581. <https://doi.org/10.1126/science.184.4136.579>.
78. Roediger, H., and Thorpe, L. (1978). The role of recall time in producing hypernesia. *Mem. Cogn.* 6, 296–305. <https://doi.org/10.3758/BF03197459>.
79. Payne, D.G. (1987). Hypernesia and reminiscence in recall: a historical and empirical review. *Psychol. Bull.* 101, 5–27. <https://doi.org/10.1037/0033-2909.101.1.5>.
80. Kapur, N. (1999). Syndromes of retrograde amnesia: a conceptual and empirical synthesis. *Psychol. Bull.* 125, 800–825. <https://doi.org/10.1037/0033-2909.125.6.800>.
81. Lewis, D.J., Misanin, J.R., and Miller, R.R. (1968). Recovery of memory following amnesia. *Nature* 220, 704–705. <https://doi.org/10.1038/220704a0>.
82. Riccio, D.C., and Richardson, R. (1984). The status of memory following experimentally induced amnesias: gone, but not forgotten. *Physiol. Psychol.* 12, 59–72. <https://doi.org/10.3758/BF03332169>.
83. Flexner, L.B., Flexner, J.B., and Roberts, R.B. (1966). Stages of memory in mice treated with acetoxycycloheximide before or immediately after learning. *Proc. Natl. Acad. Sci. USA* 56, 730–735. <https://doi.org/10.1073/pnas.56.2.730>.
84. Serota, R.G. (1971). Acetoxycycloheximide and transient amnesia in the rat. *Proc. Natl. Acad. Sci. USA* 68, 1249–1250. <https://doi.org/10.1073/pnas.68.6.1249>.
85. Squire, L.R., and Baroness, S.H. (1972). Variable decay of memory and its recovery in cycloheximide-treated mice. *Proc. Natl. Acad. Sci. USA* 69, 1416–1420. <https://doi.org/10.1073/pnas.69.6.1416>.
86. Bradley, P.M., and Galal, K.M. (1988). State-dependent recall can be induced by protein synthesis inhibition: behavioural and morphological observations. *Brain Res.* 468, 243–251. [https://doi.org/10.1016/0165-3806\(88\)90136-8](https://doi.org/10.1016/0165-3806(88)90136-8).
87. Briggs, J.F., and Olson, B.P. (2013). Reexposure to the amnesic agent alleviates cycloheximide-induced retrograde amnesia for reactivated and extinction memories. *Learn. Mem.* 20, 285–288. <https://doi.org/10.1101/lm.030270.113>.
88. Gisquet-Verrier, P., Lynch, J.F., Cutolo, P., Toledano, D., Ulmen, A., Jasnow, A.M., and Riccio, D.C. (2015). Integration of new information with active memory accounts for retrograde amnesia: a challenge to the consolidation/reconsolidation hypothesis? *J. Neurosci.* 35, 11623–11633. <https://doi.org/10.1523/JNEUROSCI.1386-15.2015>.
89. Pavlov, I. (1927). *Conditioned Reflexes* (Oxford University Press).
90. Rescorla, R.A., and Heth, C.D. (1975). Reinstatement of fear to an extinguished conditioned stimulus. *J. Exp. Psychol. Anim. Behav. Process.* 1, 88–96. <https://doi.org/10.1037/0097-7403.1.1.88>.
91. McClelland, J.L., McNaughton, B.L., and O'Reilly, R.C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol. Rev.* 102, 419–457. <https://doi.org/10.1037/0033-295X.102.3.419>.
92. O'Reilly, R.C., and Norman, K.A. (2002). Hippocampal and neocortical contributions to memory: advances in the complementary learning systems framework. *Trends Cogn. Sci.* 6, 505–510. [https://doi.org/10.1016/s1364-6613\(02\)02005-3](https://doi.org/10.1016/s1364-6613(02)02005-3).
93. Kumaran, D., Hassabis, D., and McClelland, J.L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends Cogn. Sci.* 20, 512–534. <https://doi.org/10.1016/j.tics.2016.05.004>.
94. Graham, K.S., Patterson, K., and Hodges, J.R. (1999). Episodic memory: new insights from the study of semantic dementia. *Curr. Opin. Neurobiol.* 9, 245–250. [https://doi.org/10.1016/s0959-4388\(99\)80035-x](https://doi.org/10.1016/s0959-4388(99)80035-x).
95. Staresina, B.P., Henson, R.N.A., Kriegeskorte, N., and Alink, A. (2012). Episodic reinstatement in the medial temporal lobe. *J. Neurosci.* 32, 18150–18156. <https://doi.org/10.1523/JNEUROSCI.4156-12.2012>.
96. Bosch, S.E., Jehee, J.F.M., Fernández, G., and Doeller, C.F. (2014). Reinstatement of associative memories in early visual cortex is signaled by the hippocampus. *J. Neurosci.* 34, 7493–7500. <https://doi.org/10.1523/JNEUROSCI.0805-14.2014>.
97. Tanaka, K.Z., Pevzner, A., Hamidi, A.B., Nakazawa, Y., Graham, J., and Wiltgen, B.J. (2014). Cortical representations are reinstated by the hippocampus during memory retrieval. *Neuron* 84, 347–354. <https://doi.org/10.1016/j.neuron.2014.09.037>.
98. Danker, J.F., Tompary, A., and Davachi, L. (2017). Trial-by-trial hippocampal encoding activation predicts the fidelity of cortical reinstatement during subsequent retrieval. *Cereb. Cortex* 27, 3515–3524. <https://doi.org/10.1093/cercor/bhw146>.
99. Pacheco Estefan, D., Sánchez-Fibla, M., Duff, A., Principe, A., Rocamora, R., Zhang, H., Axmacher, N., and Verschure, P.F.M.J. (2019). Coordinated representational reinstatement in the human hippocampus and lateral temporal cortex during episodic memory retrieval. *Nat. Commun.* 10, 2255. <https://doi.org/10.1038/s41467-019-09569-0>.
100. Hebscher, M., Kragel, J.E., Kahnt, T., and Voss, J.L. (2021). Enhanced reinstatement of naturalistic event memories due to hippocampal-network-targeted stimulation. *Curr. Biol.* 31, 1428–1437.e5. <https://doi.org/10.1016/j.cub.2021.01.027>.
101. Winocur, G., Frankland, P.W., Sekeres, M., Fogel, S., and Moscovitch, M. (2009). Changes in context-specificity during memory reconsolidation: selective effects of hippocampal lesions. *Learn. Mem.* 16, 722–729. <https://doi.org/10.1101/lm.1447209>.
102. Wiltgen, B.J., Zhou, M., Cai, Y., Balaji, J., Karlsson, M.G., Parivash, S.N., Li, W., and Silva, A.J. (2010). The hippocampus plays a selective role in the retrieval of detailed contextual memories. *Curr. Biol.* 20, 1336–1344. <https://doi.org/10.1016/j.cub.2010.06.068>.
103. Teyler, T.J., and DiScenna, P. (1986). The hippocampal memory indexing theory. *Behav. Neurosci.* 100, 147–154. <https://doi.org/10.1037/0735-7044.100.2.147>.
104. Teyler, T.J., and Rudy, J.W. (2007). The hippocampal indexing theory and episodic memory: updating the index. *Hippocampus* 17, 1158–1169. <https://doi.org/10.1002/hipo.20350>.
105. Goode, T.D., Tanaka, K.Z., Sahay, A., and McHugh, T.J. (2020). An integrated index: engrams, place cells, and hippocampal memory. *Neuron* 107, 805–820. <https://doi.org/10.1016/j.neuron.2020.07.011>.
106. Liu, X., Ramirez, S., Pang, P.T., Puryear, C.B., Govindarajan, A., Deisseroth, K., and Tonegawa, S. (2012). Optogenetic stimulation of a hippocampal engram activates fear memory recall. *Nature* 484, 381–385. <https://doi.org/10.1038/nature11028>.
107. Ramirez, S., Liu, X., Lin, P.A., Suh, J., Pignatelli, M., Redondo, R.L., Ryan, T.J., and Tonegawa, S. (2013). Creating a false memory in the hippocampus. *Science* 341, 387–391. <https://doi.org/10.1126/science.1239073>.
108. Vetere, G., Kenney, J.W., Tran, L.M., Xia, F., Steadman, P.E., Parkinson, J., Josselyn, S.A., and Frankland, P.W. (2017). Chemogenetic interrogation of a brain-wide fear memory network in mice. *Neuron* 94, 363–374.e4. <https://doi.org/10.1016/j.neuron.2017.03.037>.

109. Roy, D.S., Park, Y.G., Kim, M.E., Zhang, Y., Ogawa, S.K., DiNapoli, N., Gu, X., Cho, J.H., Choi, H., Kamensky, L., et al. (2022). Brain-wide mapping reveals that engrams for a single memory are distributed across multiple brain regions. *Nat. Commun.* 13, 1799. <https://doi.org/10.1038/s41467-022-29384-4>.
110. Battaglia, F.P., Benchenane, K., Sirota, A., Pennartz, C.M.A., and Wiener, S.I. (2011). The hippocampus: hub of brain network communication for memory. *Trends Cogn. Sci.* 15, 310–318. <https://doi.org/10.1016/j.tics.2011.05.008>.
111. Chetthi, S.N., Mackevicius, E.L., Hale, S., and Aronov, D. (2024). Barcoding of episodic memories in the hippocampus of a food-caching bird. *Cell* 187, 1922–1935.e20. <https://doi.org/10.1016/j.cell.2024.02.032>.
112. Chanales, A.J.H., Oza, A., Favila, S.E., and Kuhl, B.A. (2017). Overlap among spatial memories triggers repulsion of hippocampal representations. *Curr. Biol.* 27, 2307–2317.e5. <https://doi.org/10.1016/j.cub.2017.06.057>.
113. Wanja, G., Favila, S.E., Kim, G., Molitor, R.J., and Kuhl, B.A. (2021). Abrupt hippocampal remapping signals resolution of memory interference. *Nat. Commun.* 12, 4816. <https://doi.org/10.1038/s41467-021-25126-0>.
114. Rolls, E.T. (2013). The mechanisms for pattern completion and pattern separation in the hippocampus. *Front. Syst. Neurosci.* 7, 74. <https://doi.org/10.3389/fnsys.2013.00074>.
115. Brown, R., and McNeill, D. (1966). The “tip of the tongue” phenomenon. *J. Verb. Learn. Verb. Behav.* 5, 325–337. [https://doi.org/10.1016/S0022-5371\(66\)80040-3](https://doi.org/10.1016/S0022-5371(66)80040-3).
116. Brown, A.S. (1991). A review of the tip-of-the-tongue experience. *Psychol. Bull.* 109, 204–223. <https://doi.org/10.1037/0033-2909.109.2.204>.
117. Hart, J.T. (1965). Memory and the feeling-of-knowing experience. *J. Educ. Psychol.* 56, 208–216. <https://doi.org/10.1037/h0022263>.
118. Freedman, J.L., and Landauer, T.K. (1966). Retrieval of long-term memory: “Tip-of-the-tongue” phenomenon. *Psychon. Sci.* 4, 309–310. <https://doi.org/10.3758/BF03342310>.
119. Gruneberg, M.M., and Monks, J. (1974). ‘Feeling of knowing’ and cued recall. *Acta Psychol.* 38, 257–265. [https://doi.org/10.1016/0001-6918\(74\)90010-9](https://doi.org/10.1016/0001-6918(74)90010-9).
120. Reder, L.M. (1987). Strategy selection in question answering. *Cogn. Psychol.* 19, 90–138. [https://doi.org/10.1016/0010-0285\(87\)90005-3](https://doi.org/10.1016/0010-0285(87)90005-3).
121. Reder, L.M., and Ritter, F.E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *J. Exp. Psychol. Learn. Mem. Cogn.* 18, 435–451. <https://doi.org/10.1037/0278-7393.18.3.435>.
122. Schwartz, B.L., and Metcalfe, J. (1992). Cue familiarity but not target retrievability enhances feeling-of-knowing judgments. *J. Exp. Psychol. Learn. Mem. Cogn.* 18, 1074–1083. <https://doi.org/10.1037/0278-7393.18.5.1074>.
123. Koriati, A., and Lieblich, I. (1977). A study of memory pointers. *Acta Psychol.* 41, 151–164. [https://doi.org/10.1016/0001-6918\(77\)90032-4](https://doi.org/10.1016/0001-6918(77)90032-4).
124. Morton, J., Hammersley, R.H., and Bekerian, D.A. (1985). Headed records: A model for memory and its failures. *Cognition* 20, 1–23. [https://doi.org/10.1016/0010-0277\(85\)90002-2](https://doi.org/10.1016/0010-0277(85)90002-2).
125. Ruchkin, D.S., Grafman, J., Cameron, K., and Berndt, R.S. (2003). Working memory retention systems: A state of activated long-term memory. *Behav. Brain Sci.* 26, 709–728. <https://doi.org/10.1017/s0140525x03000165>.
126. Norris, D. (2017). Short-term memory and long-term memory are still different. *Psychol. Bull.* 143, 992–1009. <https://doi.org/10.1037/bul0000108>.
127. Ball, F., and Busch, N.A. (2015). Change detection on a hunch: Pre-attentive vision allows “sensing” of unique feature changes. *Atten. Percept. Psychophys.* 77, 2570–2588. <https://doi.org/10.3758/s13414-015-0963-9>.
128. Mandler, G. (1980). Recognizing: the judgment of previous occurrence. *Psychol. Rev.* 87, 252–271. <https://doi.org/10.1037/0033-295X.87.3.252>.
129. Ryan, T.J., Roy, D.S., Pignatelli, M., Arons, A., and Tonegawa, S. (2015). Engram cells retain memory under retrograde amnesia. *Science* 348, 1007–1013. <https://doi.org/10.1126/science.aaa5542>.
130. Roy, D.S., Muralidhar, S., Smith, L.M., and Tonegawa, S. (2017). Silent memory engrams as the basis for retrograde amnesia. *Proc. Natl. Acad. Sci. USA* 114, E9972–E9979. <https://doi.org/10.1073/pnas.1714248114>.
131. LeCun, Y., Cortes, C., and Burges, C.J. (1998). The MNIST database of handwritten digits. <https://oss-ci-datasets.s3.amazonaws.com/mnist>.
132. Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1708.07747>.
133. Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychol. Rev.* 97, 285–308. <https://doi.org/10.1037/0033-295x.97.2.285>.
134. French, R.M. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 3, 128–135. [https://doi.org/10.1016/s1364-6613\(99\)01294-2](https://doi.org/10.1016/s1364-6613(99)01294-2).
135. Gershman, S.J. (2024). What have we learned about artificial intelligence from studying the brain? *Biol. Cybern.* 118, 1–5. <https://doi.org/10.1007/s00422-024-00983-2>.
136. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimeshe, N., Antiga, L., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems (NeurIPS)*, 8026–8037.
137. Kinga, D., and Adam, J.B. (2015). A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 5, p. 6.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
MNIST	LeCun et al. ¹³¹	https://ossci-datasets.s3.amazonaws.com/mnist
Fashion MNIST	Xiao et al. ¹³²	http://fashion-mnist.s3-website.eu-central-1.amazonaws.com
Software and algorithms		
PyTorch 2.5.1	Paszke et al. ¹³⁶	https://pytorch.org
Code for this paper	This paper	https://doi.org/10.5281/zenodo.14920916

METHOD DETAILS

Our code was implemented using PyTorch.¹³⁶

Distinct representations for keys and values

In both the two-class and three-class settings, 100 two-dimensional vectors were uniformly sampled from the range [0, 1] for each key and value within each class, resulting in a total of 200 key/value pairs for the two-class case (Figure 2A) and 300 for the three-class case (Figure 2B).

The optimization was performed using the Adam optimizer¹³⁷ with a learning rate of 3e-4 for 5,000 steps in the two-class setting and 10,000 steps in the three-class setting.

The experiments were conducted using the free version of Google Colab with a CPU. The python notebooks to reproduce the results are https://github.com/kazuki-irie/kv-memory-brain/blob/master/2classes_key_value_optimization.ipynb and https://github.com/kazuki-irie/kv-memory-brain/blob/master/3classes_key_value_optimization.ipynb for the two-class and three-class cases, respectively.

Forgetting as retrieval failure, and recovery by memory reactivation

Both the MNIST and Fashion MNIST datasets were used without modifications. We used the class ‘0’ and ‘1’ images from each dataset; resulting in 12,665 training images for MNIST and 12,000 for Fashion MNIST. Note that MNIST does not have an equal number of examples per class: 5,923 for class ‘0’ and 6,742 for class ‘1.’ We left this imbalance as is, as it is irrelevant to the main goal of our experiment. The corresponding test sets consist of 2,115 and 2,000 images for MNIST and Fashion MNIST, respectively.

The initial weights of the two linear layers within the feedforward neural network were uniformly sampled from a uniform distribution over the range $[-a, a]$ where $a = 1/\sqrt{d_{in}}$, and d_{in} denotes the input dimension of the corresponding layer. No bias was applied in either of the two layers. The optimization was conducted using the standard stochastic gradient descent algorithm using a learning rate of 6e-4 and a batch size of 128 images. We refer to one “epoch” of optimization as a single iteration over the entire training dataset.

The experiments were conducted using the free version of Google Colab with a T4 GPU. The python notebook to reproduce the corresponding results can be found at: https://github.com/kazuki-irie/kv-memory-brain/blob/master/Forgetting_and_recovery.ipynb.