Associative memory of structured knowledge

Julia Steinberg[1,2,3,*] and Haim Sompolinsky[2,4,†]

[1] *Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA*
[2] *Center for Brain Science, Harvard University, Cambridge MA 02138, USA*
[3] *Joseph Henry Laboratories of Physics, Princeton University, Princeton NJ 08544, USA*
[4] *Edmond and Lily Safra Center for Brain Sciences, Hebrew University, Jerusalem 91904, Israel*

A long standing challenge in biological and artificial intelligence is to understand how new knowledge can be constructed from known building blocks in a way that is amenable for computation by neuronal circuits. Here we focus on the task of storage and recall of structured knowledge in long-term memory. Specifically, we ask how recurrent neuronal networks can store and retrieve *multiple* knowledge structures. We model *each* structure as a set of binary relations between events and attributes (attributes may represent e.g., temporal order, spatial location, role in semantic structure), and map each structure to a distributed neuronal activity pattern using a vector symbolic architecture (VSA) scheme.

We then use associative memory plasticity rules to store the binarized patterns as fixed points in a recurrent network. By a combination of signal-to-noise analysis and numerical simulations, we demonstrate that our model allows for efficient storage of these knowledge structures, such that the memorized structures as well as their individual building blocks (e.g., events and attributes) can be subsequently retrieved from partial retrieving cues. We show that long-term memory of structured knowledge relies on a new principle of computation beyond the memory basins. Finally, we show that our model can be extended to store sequences of memories as single attractors.

Human memory is remarkable in its ability to robustly store and retrieve information with complex and hierarchical structure, guiding cognitive processes on many different timescales. In many instances, this "structured knowledge" can be described as sets of associations between discrete events with their contextual attributes. Some concrete examples are, temporal sequences representing events associated to particular times, episodic memories representing events associated with particular contexts [1, 2], cognitive maps representing spatial environments through landmarks associated with locations [3–5], and semantic structures in language in which meaning is conveyed through sets of words associated with their respective roles within a sentence [6–8].

To effectively use structured knowledge that has been stored in long-term memory, it must be represented in a way that allows for its retrieval through partial information, with tolerance for noisy and degraded cues. This is likely facilitated by the distributed nature of the underlying neural representations, which provide an inherent notion of similarity between representations and a mechanism for learning representations by the the tuning of synaptic weights in a neural network [7, 9–11]. However, while the utility of distributed representations is clearly beneficial from this perspective, it is still not well understood how to represent associations and relations in neural networks in an efficient and flexible way that is amenable to the variety of computational demands involved in higher cognition [12–14].

Several recent studies have addressed the contextual modulation of neuronal representations, e.g., by forming "mixed representations" [15] or by gating parts of the network [16]. Other proposals have tried to implement more general relational structures in neural networks. An early attempt used the tensor product to create a distributed representation of pairwise relations between discrete items [6]. Subsequently, several Vector-Symbolic Architectures (VSA) were proposed as compressions of the tensor product to avoid the increase in dimensionality of the representation, allowing for the creation of hierarchies of relations in a compact way [17–22]. More recently, several architectures for deep or recurrent neural networks have been proposed to promote flexible relational reasoning [23–29]. However, all of these works have primarily focused on working memory, i.e., online tasks of processing incoming structured data. By contrast, the challenge of storing and retrieving relational structures in long-term memory has received little attention.

Storing knowledge structures in long-term memory poses several additional challenges. While working memory tasks typically process one structure at a time, long-term memory networks must cope with storing multiple structures, such as multiple complex cognitive maps, multiple temporal sequences, or multiple stories. Two generic measures of the efficiency of information storage in recurrent neural networks are their extensive capacity, i.e., the number of stored items scales with the number of neurons in the network [30], and the ability to recall memories from partial cues which have small but significant overlap with the desired memory [31]. Both of these measures can be adversely affected by correlations across memorized patterns. For relational structures, additional correlations may occur due to the presence of objects or contextual attributes in multiple memories, putting additional constraints on the encoding of relational information. In addition to these considerations, models of

* jsteinberg@princeton.edu
† haim@fiz.huji.ac.il

distributed representation of knowledge structures typically compress the relational structure into a fixed-length distributed vector representation. To compensate for this loss of information, "clean-up" mechanisms are invoked. Thus, it is crucial that such mechanisms can be adapted for the task of retrieval of such structures from long-term memory to efficiently store large numbers of relational structures each containing multiple associations.

In this work, we propose a model for associative memory of *multiple* relational structures by using a quadratic binding scheme to form vector representations of memories consisting of *multiple* binary relations between items (which we will henceforth denote as pairs of objects and their attributes). While our model is quite general, in most of our work we will use the holographic reduced representation (HRR)[7] VSA scheme for convenience. We show that the binarized versions of these structures can be stored as *fixed point* attractors in a recurrent neural network and each structure can be retrieved from the memory network by using a cue which is a structure encoding a subset of the relations in the memorized structure. We highlight the holistic nature of this model by comparing the storage of temporal sequences in the present model, where the entire sequence is stored as a single fixed point, to previous models, where a sequence is stored as a sequence of transitions between multiple fixed points and cannot be fully recalled at once [32]. Our model posits that in addition to the network that stores the structures, a Dictionary network stores all individual items (e.g., individual words, familiar objects). We show that the identities of the objects contained in the structure can be decoded faithfully from the retrieved memory by querying the retrieved structure with the appropriate cue as long as a "clean-up" operation is performed to map the noisy estimate of the object to the correct item in the Dictionary. Furthermore, this decoding works well even when the retrieved structure is significantly degraded.

RELATIONAL STRUCTURES

We begin by modeling a binary relational structure S as a set of L object/attributes pairs

$$S = \{(a_1, b_1), \ldots, (a_L, b_L)\} \tag{1}$$

where both objects a and attributes b have embeddings as real vectors representing distributed patterns of activation in a neuronal population. For simplicity, both populations will be assumed to be of the same size N. We represent relations between items in a pair (a, b) by a transformation through a pairwise quadratic nonlinearity to a binding vector $g(a, b)$ (in $\mathbb{R}^N$) representing a distributed pattern of activity in a population of N "binding" neurons. Each component of the binding vector takes the form

$$g_k(a, b) = a^T G^k b \tag{2}$$

where each G^k is an $N \times N$ fixed binding matrix. The binding operation in Eqn. 2 is a generalized version of a VSA scheme [18] and can be interpreted as a lossy compression of the tensor product binding operation first proposed in [6].

We obtain the representation of the full relational structure S by the vector summation of the individual object/attribute pairs,

$$\widehat{S} = \sum_{\ell=1}^{L} g(a_\ell, b_\ell) \tag{3}$$

where the vector summation induces a second source of information loss. The representation $\widehat{S}$ is permutation invariant with respect to the index ℓ so that the relations within the structure have no particular order.

The compressed representations of structures can be used for a variety of computations, such as structure classification. Here we focus on unbinding tasks, in which given $\widehat{S}$ and one of its attributes b_ℓ, we need to estimate its pair a_ℓ. Similar to binding, we assume that the unbinding operation is performed through a quadratic transformation involving the pair $(\widehat{S}, b)$, so that the k-th component of the estimator $\hat{a}_\ell^k$ of a_ℓ is given by

$$\hat{a}_\ell^k = \widehat{S}^T F^k b \,,\; k = 1, ..., N \tag{4}$$

where each F^k is an $N \times N$ matrix chosen so that the decoding operation is the approximate inverse of the binding operation.

In general, the binding and unbinding matrices can be learned and the optimal choice should depend on the nature of the items contained in the dictionary. Here we use a generic set of matrices, a popular choice known as Holographic Reduced Representations (HRR) described in Methods.

The final estimate of a, $\tilde{a}$, is computed by comparing the noisy estimate against a Dictionary, i.e., a neural long-term memory system that stores all familiar objects a_d, using,

$$\tilde{a} = \arg\max_{d \in \mathcal{D}} d \cdot \hat{a} \tag{5}$$

A schematic of the encoding and decoding networks is shown in Fig. 1a.

The maximum likelihood (ML) decoding error is given by the probability P_ϵ that the estimator $\hat{a}$ has the largest overlap with an incorrect item in the dictionary. It depends on the size of the dictionary D and the signal-to-noise ratio (SNR) overlaps of the estimator $\hat{a}$ with items dictionary defined in Eqn. 20 in Methods. ML decoding is an idealization of a more biologically realistic retrieval of the stored pattern a from a retrieval cue $\hat{a}$ in a long-term associative memory network storing all individual Dictionary items. Two possible implementations of this memory system are a winner take all network with lateral inhibition [33] or a sparse Hopfield network [34].

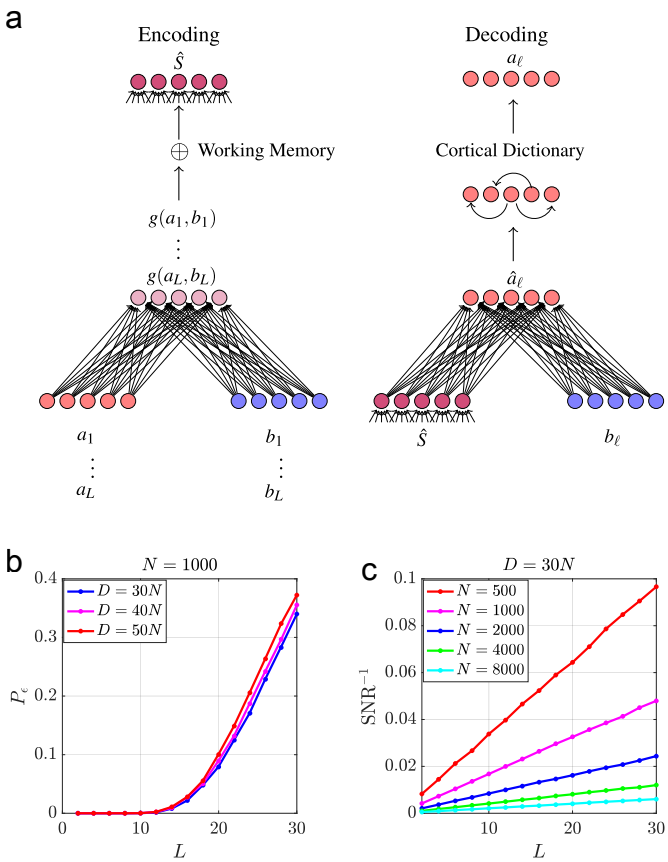

FIG. 1. (a) A schematic of the network used to bind objects/attribute pairs (a, b) to form the knowledge structure $\widehat{S}$ alongside the network used to decode an object a_ℓ from $\widehat{S}^\mu$ by presenting the attribute b_ℓ. (b) The unbinding error P_ϵ shown as a function of structure length L for $N = 1000$ for various values of D averaged $10,000$ structures (C) SNR^{-1} shown as a function of L for several values of N averaged over $10,000$ structures.

STORING STRUCTURES IN LONG-TERM ASSOCIATIVE MEMORY

We now consider the long-term memorization of multiple knowledge structures by storing their vector representations in a neural network for long-term Structured Knowledge Associative Memory (SKAM), so that they can be retrieved at a later time from partial cues and subsequently queried to reconstruct individual events. We consider a set of P structures $\{S^1, \ldots, S^P\}$ which for simplicity, all consist of L items. We label the set of L objects and attributes comprising the μ-th structure as a^μ_ℓ, $b^\mu_\ell, l = 1, ..., L$. For each structure, the HRR encoding scheme is used to create a vector representation $\widehat{S}^\mu$ from S^μ.

To store multiple structures as fixed points in a neural network, the neuronal input-output transformation must be highly non-linear, implying that the stored patterns themselves are limited to the dynamic range of the neurons. As in a standard Hopfield network [30, 31, 35], we assume neurons are binary ± 1 variables and the memory patterns, candidates of fixed points of the attractor dynamics are $\sigma^\mu = \text{sgn}(\widehat{S}^\mu)$. For a network with N binary neurons, the memory load is defined as $\alpha = \frac{P}{N}$, where P is the number of stored structures.

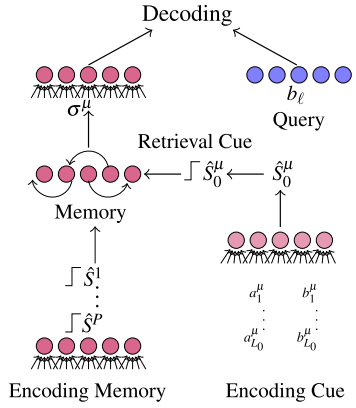

FIG. 2. A schematic of the process of storing multiple binarized structures in a memory network. These structures can be retrieved from memory by encoding a retrieval cue from a subset of the relations in the desired structure as in Eqn. 6 and initializing the network in the state of the binarized cue.

In general, the associative nature of Hopfield memory networks is expressed in the ability to recall a memorized pattern starting in any initial state which has a sufficiently large overlap with the memorized pattern. If the initial state is within the basin of attraction of the pattern, it will converge to the pattern without errors. In our case, we consider partial cues of a structure, $\widehat{S}$, in the form of a recalling structure $\widehat{S}_0$ obtained by the binding and subsequent summation of any subset S_0 of the L pairs of binary relations contained in $\widehat{S}$, i.e.,

$$S_0^\mu = \sum_{\ell=1}^{L_0} g(a^\mu_\ell, b^\mu_\ell) \tag{6}$$

where L_0 is the number of recalling elements in $\widehat{S}_0$ and is assumed to be much less than L. The network is then initialized in the state $\sigma_0^\mu = \text{sgn}(\widehat{S}_0^\mu)$ and evolved to a fixed point which, if successful, corresponds to the stored binarized structure σ^μ. A schematic of our model is shown in Fig. 2 and more details on the memory network are given in Methods.

There are several learning rules which can be used to store patterns as discrete fixed points in recurrent neural network models of associative memory. Here, for simplicity, we use the Pseudo-inverse learning rule proposed in [36] to train the network. In a Pseudo-inverse network, all structures are perfect fixed points for $\alpha < 1$, which is assumed throughout. This allows us to focus on the retrieval cues, since failure to perfectly recall a structure occurs only when the retrieval cue is outside of

the basin of attraction of the memorized structure. We observe qualitatively similar behavior for the Hebb learning rule and the Storkey rule introduced in [37] for α well below the memory capacity described in Supplementary Appendix III D.

RESULTS

We evaluate the performance of the scheme introduced in the previous section by the ability to accurately perform the unbinding operation after retrieval of a structure from the SKAM. For example, after retrieving the structure $\mu = 1$, we should be able to extract the item a_ℓ^1 with a query in the form of its pair, i.e., b_ℓ^1 with low error. We quantify performance by the average unbinding error P_ϵ obtained in simulations where structured memories are created from random patterns, stored in memory, retrieved with partial cues $\sigma_0^\mu = \mathrm{sgn}(\widehat{S}_0^\mu)$, and subsequently decoded using the ML "clean-up" operation. We assume all items appearing in memorized structures are stored in dictionaries for objects and for attributes which are then used to decode from the retrieved memory. A schematic of this process is shown in Fig. 2 and full details of the simulations are provided in Methods.

The parameters involved in the performance measure are: network size N, memory load $\alpha = P/N$, size of the relational structures and the retrieval cue, denoted as L and L_0, respectively. We consider the regime where both N and P are very large and the memory load $\alpha \sim O(1)$ [30, 38], mainly considering values $\alpha \sim 0.1 - 0.2$, where the network acts as a good associative memory.

Retrieval of Structured Memories

We begin by showing numerical results which measure the quality of the retrieved structures in terms of the unbinding error P_ϵ and the SNR of overlaps defined in Eqn. 20. In all reported results, the extracted item (and the associated query) comes from pairs that are not part of the cueing structure S_0. Thus any performance better than chance necessarily involves information extracted by the retrieval from long-term memory. In Fig. 3a we show the dependence of the unbinding error P_ϵ on L, L_0, and α. For comparison, we show P_ϵ for the original structure prior to storage in the memory network, demonstrating that except for small L, the dominant contribution to the error comes from retrieving the structure from long-term memory. We also observe that for a fixed L, L_0, and α, the error is suppressed as N increases, in contrast to standard large attractor memory networks where performance depends only on P/N. To elucidate this behavior, we replot the results in terms of the SNR^{-1} , i.e., the inverse of the SNR as defined above in Fig. 3b, showing that for each L_0, there is a critical L above which the SNR of the memorized structures decreases relative to the original SNR, SNR_0 before storage in long-term memory. Note that due to binarization, SNR_0 is smaller by a factor of $\frac{2}{\pi}$ relative to the value given in Eqn. 22. Finally, in Fig. 3c we replot the same results in terms of the inverse of the normalized SNR, $\mathrm{SNR}/\mathrm{SNR}_0$ vs., L/L_0. Since SNR_0 is proportional to N/L, this normalization factors out the "trivial" dependence on L/N from the post retrieval SNR. Fig. 3c shows that for a fixed α the normalized inverse SNR depends only on L/L_0. and only weakly on N, suggesting that the main N dependence comes from the linearity of SNR_0 in N.

Length of Cueing Structure and Memory Basins

As seen in Fig. 3, the performance worsens (and SNR decreases) as L increases, while the converse holds true for L_0. We find that there is a critical ratio, $l_c = \frac{L_0}{L}$, defined as the minimum initial cue (relative to the total length), that leads to very small error, essentially equivalent to the error for the original structure.

To understand the origin of l_c, we note that $\frac{L}{L_0}$ specifies the average initial overlap of the retrieval cue with the corresponding memorized structure, which we denote m_0. For small values of L_0, $m_0 \approx \frac{2}{\pi}\sqrt{\frac{L_0}{L}}$. The size of m_0 determines whether on average the initial state is within the basin of the desired memory, so that the recurrent dynamics will succeed (or fail) in converging to the desired attractor. As the cueing length L_0 grows, the initial state becomes increasingly likely to be within the basin of attraction of the desired structure, retrieving it with essentially no error. In these conditions, the unbinding operation has the same probability of success as for the original structure. Conversely, for small enough L_0 the initial state is likely outside the attraction basin of the memory, hence the retrieval will be noisy.

To determine the minimum L_0 required for perfect retrieval, we use known estimates [36] of the radius of attraction in attractor memory networks, $R(\alpha) = 1 - m_{min}(\alpha)$, where $m_{min}(\alpha)$ is the minimal overlap between the initial state and the desired memory required for convergence to the correct fixed point on average . $m_{min}(\alpha)$ determines the minimal length of the cueing structure, i.e., $l_c(\alpha) \approx \frac{\pi}{2} m_{min}^2(\alpha)$.

We conclude that when $L_0/L < l_c$, the main source of the decoding error in our model comes from the limitation on good retrieval of the structure from memory, due to small values of L_0/L, and not from noise in the original encoding (corresponding to $L_0/L > l_c$).

Retrieval Outside Memory Basins

Naively, one would expect that for $L_0 < l_c L$, P_ϵ will be very large due to the accumulation of errors in the retrieved structure, which is outside the memory basin. However, as shown in Fig. 3, this is not case. Surprisingly, the decoding performance is well below chance level for

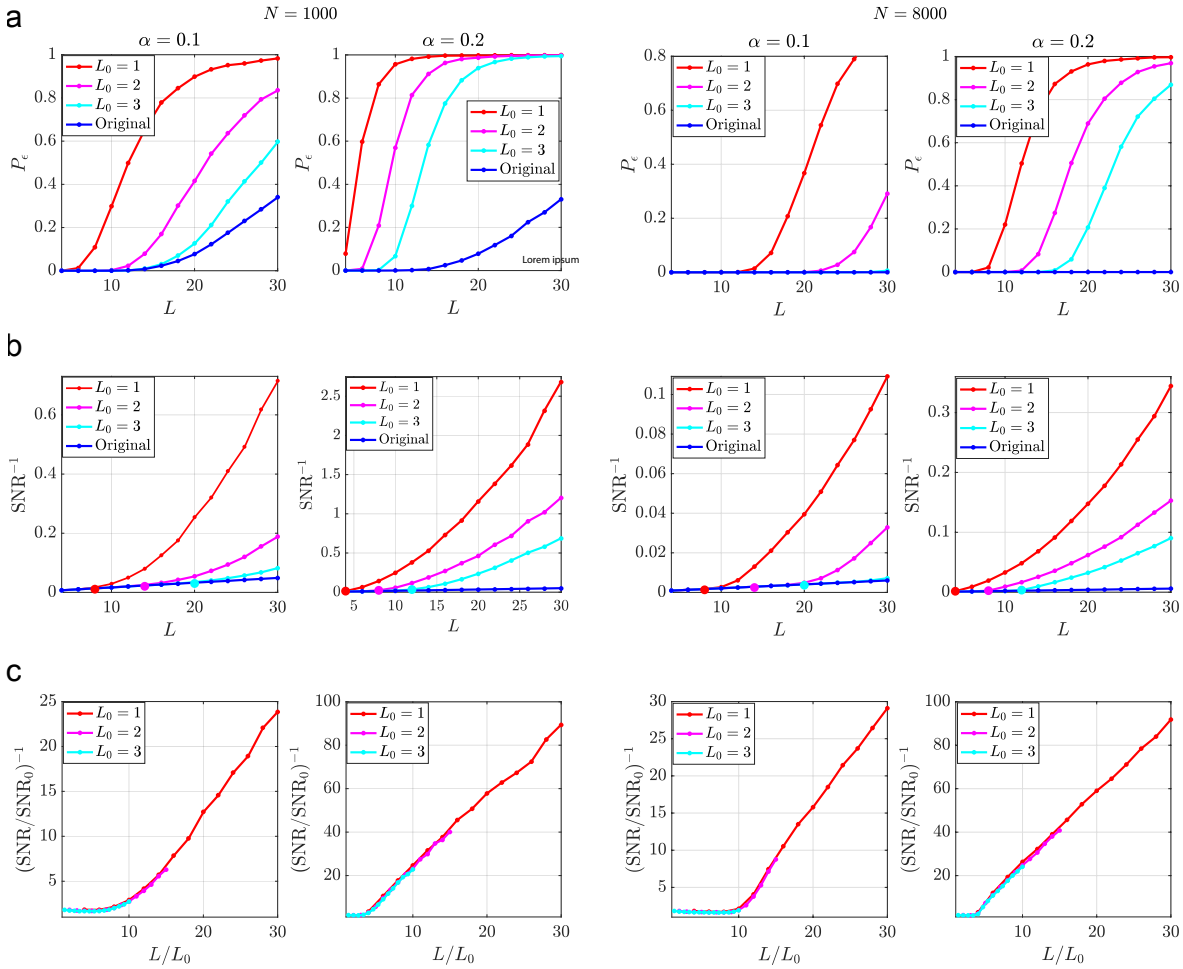

FIG. 3. (a) The decoding error P_ϵ is compared for several values of L_0 from structures containing L pairs of items for two values of N and two values of α. (b) SNR^{-1} v. L. For each value of L_0, l_c is given by the value of L where SNR^{-1} diverges from the line corresponding to the original structure, which is marked for each value of L_0. (c) $(\mathrm{SNR}/\mathrm{SNR}_0)^{-1}$ v. $\frac{L}{L_0}$ where $\mathrm{SNR}_0 = \frac{N}{L}$. For all figures, $T = 20$ parallel updates are used in memory retrieval and the average is performed over $10,000$ memories. The dictionary is fixed to $D = 30N$.

substantial range of values of L, even when $L \gg L_0/l_c$. This observation can be explained by two scenarios: (1) the actual basins fluctuate in their shape so that for some structured memories, initial states may converge to the memory fixed point even if they are outside the *mean* basin radius; (2) initial states outside the true memory basin converge to fixed points with significant overlap with the structured memory.

To test these scenarios, we measured the empirical distributions $p(m)$ where m is the overlap between the fixed point and the desired structure, obtained from histograms of overlaps for several values of $L > \frac{L_0}{l_c}$, shown in Supplementary Fig. 3. We find that as N is increased, $p(m)$ becomes sharply peaked around a single value m^*. Inside the basin of attraction, i.e., $m_0 < m_{min}(\alpha)$, $m^* = 1$. However, outside of the basin when $m_0 < m_{min}(\alpha)$, $m^* < 1$; nevertheless it is substantially larger than 0. The value of m^* depends on both m_0 and the load α roughly as

$$m^*(\alpha, m_0) \sim f(\alpha) m_0 \tag{7}$$

described in further detail in Methods and the Supplementary Appendix. III. Furthermore, for $\alpha \lesssim 0.3$, $f(\alpha) > 1$ (Fig. 4d), implying that the final overlap with the retrieved structure is significantly larger than the initial overlap m_0 even far outside the basin for the structure.

SNR of Retrieved Structures Outside the Basin

We now use the preceding results to estimate the SNR for $L_0/L \ll l_c$, i.e., when the initial state is well outside the memory basin. First, we argue that the SNR of unbinding from a noisy state with overlap $m < 1$ with the

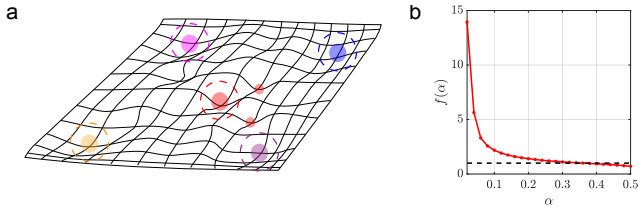

FIG. 4. (a) Schematics of the energy landscape of the memory network. The large filled circles are stored memories and the dashed circles denote their basins of attraction. The two small red circles are fixed points outside of the basin of attraction of the red memory which still lead to a large enough overlap for accurate decoding when $1 \gg N/L$. (b) $f(\alpha)$ defined in Eqn. 7 is obtained from simulations of a Pseudo-inverse network storing random memories of size $N = 8000$. The black dotted line at 1 shows that $m^* > m_0$ for $\alpha \lesssim 0.3$. $T = 20$ parallel updates are used for memory retrieval and averages are performed over 50 trials.

true structure, should be roughly,

$$\mathrm{SNR}(m) \sim \frac{2c}{\pi}\frac{m^2 N}{L} \tag{8}$$

where, as before, the factor of $\frac{2}{\pi}$ comes from binarization and $c \approx 0.65$ accounts for the fact that part of the overlap m is contributed by the initial cueing structure S_0 and is more concentrated around the relations contained in the retrieval cue. For very large networks, we can replace m in Eqn. 8 with m^* from Eqn. 7. Expressing m_0 in terms of L_0/L by using Eqn. 30 from Methods, we arrive at

$$\mathrm{SNR} \sim \frac{8cf(\alpha)^2}{\pi^3}\frac{NL_0}{L^2},\ L_0 \ll L \tag{9}$$

verified in Fig. 5 for two values of α. These results summarize the rich behavior of associative memory of structured knowledge. In contrast to standard memory functions, here the performance depends not only on the memory α but also on the network size N, structure length L, and cueing length L_0 through the SNR. The key difference is that in structured memories, the criterion for success is not limited to convergence to the target memory; even if the target memory is only partially retrieved, the underlying memorized relations can be still be retrieved faithfully using the semantic memory. The well-known property of pattern completion is realized here by a sub-structure of length $L < L_0$, in addition to the standard random initial condition.

STORAGE AND RETRIEVAL OF SEQUENCES

Storing Sequences as Binary Structures

We now extend the results of the previous sections to representations of temporal sequences. Temporal sequences can be modelled as structures in several ways.

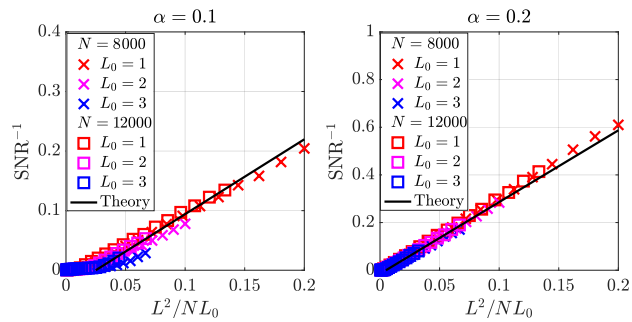

FIG. 5. SNR^{-1}v. L^2/NL_0 shown for $N = 8000$ and $N = 12000$ and for several values L_0 as an average over items contained in 10,000 memories shown for $\alpha = 0.1$ and $\alpha = 0.2$. The black line is obtained from Eqn. 9 with $c = 0.65$. $T = 20$ parallel updates are used for memory.

One possibility is to bind each event in the sequence with its temporal order in the sequence. This can be implemented via a contextual drift process with a context representation that evolves as items in the sequence are retrieved as in the Temporal Context Model of free recall of lists [2] and similarly in the Context Retrieval and Updating model [39, 40]. Here we use temporal proximity as the contextual cue by interpreting a sequence as a set of binary associations between temporally proximal events. Thus, a temporal sequence of length L, $(a_1, a_2, ..., a_L)$ can be represented as a structure of the form

$$S = \{(a_1, a_2), (a_2, a_3), \ldots, (a_{L-1}, a_L)\} \tag{10}$$

and the entire sequence S is represented by a vector $\widehat{S}$ of size N given by

$$\widehat{S} = \sum_{\ell=1}^{L-1} g(a_\ell, a_{\ell+1}) \tag{11}$$

Decoding an episode at a particular time, i.e., a_ℓ, is performed through an unbinding operation with a query by the preceding event, $a_{\ell-1}$. Starting from a query by a_1, the entire sequence can be unfolded through a sequence of queries. Because each event appears in two binary relations, we need to use an asymmetric binding operation so that $g(a, b) \neq g(b, a)$. Within HRR, this can be accomplished by switching the binding and unbinding operations [41].

As before, we consider the case in which all items being decoded are contained in a Dictionary $\mathcal{D} = \{a_1, a_2, \ldots, a_D\}$, so each decoding step involves a clean-up of the decoded item before preceding to decode the next item from the sequence. A schematic of this process is shown in Fig. 6a.

The binarized versions of the structures representing each sequence are stored for long-term memory in a recurrent neural network with synaptic weight matrix determined via the Pseudo-inverse rule. The cueing structure $\widehat{S}_0$ consists of the first relation (a_1, a_2), so the overlap of

$\widehat{S}_0$ with the stored sequence $\widehat{S}$ is closely approximated by Eqn. 30 with $L_0 = 2$. Alternatively, the first item a_1 can be used as a retrieval cue if it is added to the representation $\widehat{S}$ in Eqn. 11. We are primarily interested in the ability to reconstruct the entire sequence after it is retrieved from memory.

Retrieval of Sequences from Long-term Memory

Because of the sequential nature of the decoding of sequences in our model, the decoding error is accumulating and performance of unbinding of an event depends on its position in the sequence (relative to the cued events). In Fig. 6B, we show the decoding error P_ϵ at each position along sequences encoded by Eqn. 11 for sequences of different length L. Since the SNR of the overlap with the correct item along each position in the sequence depends on L, the length of the sequence limits the accuracy of decoding at all positions along the sequence. Nevertheless, for low memory loads and moderately long sequences, the accumulated error is small.

Decoding Error

An interesting outcome of this mode of recall is the accumulation of errors as the recall sequence advances. This would give rise to correlations between probabilities of recall that decay as a function of the temporal lag between the events, consistent with observations [2]. This behavior was previously explained by positing that proximal temporal context vectors are correlated. In our model, these correlations are a natural consequence of the fact that the proximal events serve as temporal context cues.

The present scheme of long-term storage of sequences as single fixed points overcomes a key disadvantage of previous attempts at storing multiple temporal associations in attractor neural networks. In previous models of sequential memory [32], all of the patterns contained in the sequences are stored as separate attractors in one network and the sequences themselves are encoded in a time-delayed synaptic plasticity rule that associates each pattern with its next pattern in the sequence, illustrated in Fig. 6c and reviewed in Supplementary Appendix IV. Because of the Markovian nature of the synaptic plasticity, retrieval will fail if multiple sequences share the same event, as illustrated in Fig. 6d. By contrast, in the present model, the entire sequences are stored and retrieved as separate "holistic" attractors. Thus, as long as the retrieval cue is unique to a single sequence, it will retrieve it unambiguously and the subsequent unfolding of the sequence by the unbinding network will be immune from interference with other sequences. To demonstrate this, we consider P sequences S^μ with $\mu = 1, \ldots, P$ of length L where neighboring sequences S^μ and $S^{\mu+1}$ share L_s events in common. An example of storing two sequences with an overlapping event is shown in Fig. 6e demonstrating the successful retrieval of the entire sequence despite the presence of an overlapping state (compare with Fig. 6d). Figure 6f shows that sequences can

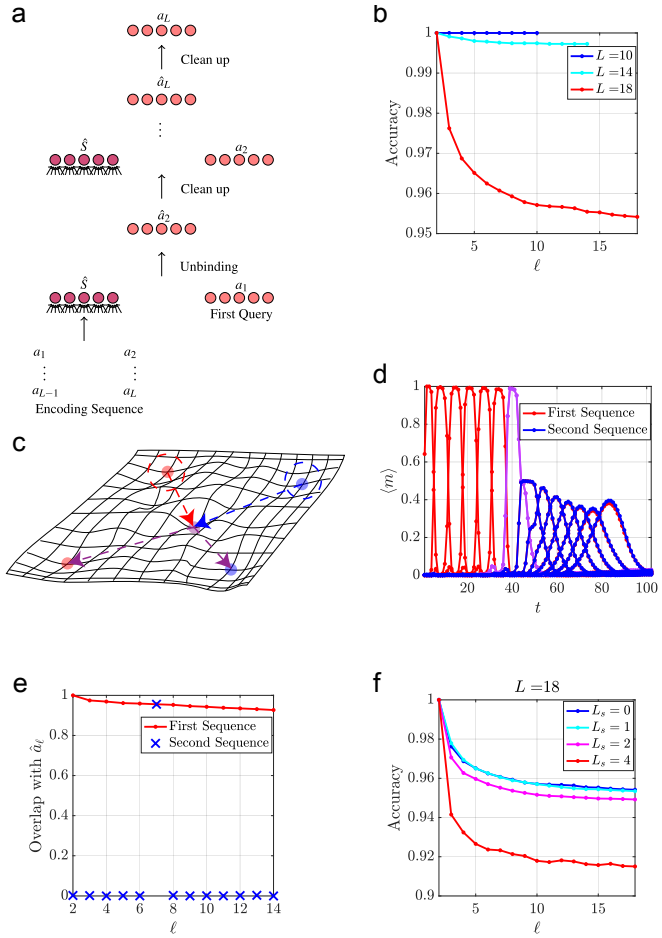

FIG. 6. (a) A schematic of the decoding scheme for sequences. (b) Decoding accuracy as a function of sequence position ℓ along sequences of length L retrieved from memory shown for several values of L. $N = 2500$, $\alpha = 0.05$, and the average is computed over 100 trials. (c) A schematic of the energy landscape for two temporal sequences containing a single common element which are encoded as sequences of attractors using the scheme in [32] (discussed in Supplementary Appendix IV). (e) The average overlap of the network state $\langle m \rangle$ with each attractor in the sequence is shown as a function of retrieval time for a network of size $N = 1000$ storing two sequences containing a single element in common. The parameters $\tau = 8$ and $\lambda = 2.5$ are used and the average is computed over 1000 trials. (e) The average overlap of the estimator $\hat{a}_\ell$ (normalized by $\langle \hat{a}_2 \cdot a_2 \rangle$) with the correct Dictionary item at each position in the sequence for a network of size $N = 1000$ storing two sequences containing a single element in common. The average is computed over $10,000$ trials. (f) Decoding accuracy as a function of sequence position ℓ for $P = \alpha N$ sequences for which each neighboring sequence contain L_s common elements with the previous one. $N = 2500$, $\alpha = 0.05$ and the average is computed over 100 trials.

be faithfully retrieved even with multiple common states, up to the point where the basins of attraction of individual sequences shrink due to the large overlap between them.

Finally, it is interesting to compare the the memory capacity of the two models. In the sequence attractor model, the maximal number of stored sequences of length L is $P < \alpha_c N/L$ since the network stores PL states. In contrast, in the present model, since only P attractors are stored, the capacity of storage is $P < \alpha_c N$. Nevertheless, for a successful unfolding of the sequence we need PL/N to be bounded (for a fixed L_0/L) due to the noise in the unbinding operation. A potential disadvantage of the current model is the need to devote additional memory resources to store the individual events in a Dictionary. On the other hand, the Dictionary can be used for multiple other cognitive tasks aside from the retrieval of these sequences. Another flexibility in the separation of the retrieval of the neural representation of the sequence from the subsequent reconstruction of individual events, is the fact that, for some tasks, the agent may not need to access the full detail of the sequence, for instance in tasks that requires distinguishing between one episode and another one. Such tasks may not need to rely on the full unbinding of the sequences.

NEURAL IMPLEMENTATION OF MULTIPLICATIVE BINDING

We now briefly consider possible implementations of the binding computation in Eqn. 2 through multiplicative interactions in biological neurons [42]. Previously, several mechanisms have been proposed to facilitate multiplicative interactions among neurons including dendritic gating [43], quadratic firing rate response [44], and short-term synaptic plasticity. Short-term plasticity comprises a variety of synaptic processes that modulate synaptic efficacy in a transient, activity-dependent manner [45]. These processes occur on timescales ranging from milliseconds to minutes [46] and are thought to mediate higher cognitive functions like attention and working memory. More recently it has been suggested that "fast weights" in artificial neural networks may serve as an analogy to short-term plasticity in the brain [47] with connections to linear transformers [48].

We start by noting that $g_k(a,b) = a^T b'$ where $b' = G^k b$. The last term is a representation of the activity pattern b by propagating it through a synaptic matrix G^k. Finally the dot product between a and b' can be implemented can be decomposed into an outer product of two fixed synaptic weight vectors, i.e. $G^k = w_a^k w_b^{kT}$ so that components of the binding vector take the form

$$g_k(a,b) = (w_a^k \cdot a)(w_b^k \cdot b) \tag{12}$$

We now use the above form to consider how firing rate nonlinearity and short-term synaptic plasticity can serve as mechanism for generating quadratic binding.

Nonlinearity of the Firing Rate

Biological neurons can potentially implement the computation of the binding vector $g(a,b)$ via the nonlinearity of the firing rate response to a synaptic current $r = f(I)$ where the synaptic current is given by the sum $I = w_a^k \cdot a + w_b^k \cdot b$. While for many neurons rectificed nonlinearity $f(I) = [I - I_0]_+$ is a good approximations, other neurons are found to be approximated by quadratic nonlinearity $f(I) = [I - I_0]_+^2$ where the firing in response to the sum of the separate responses to each input a and b is subtracted from the response to the combined input from a and b. Building on a quadratic $f(I)$ curve, we can write

$$f(w_a^k \cdot a + w_b^k \cdot b) - f(w_a^k \cdot a) - f(w_b^k \cdot b) = 2(w_a^k \cdot a)(w_b^k \cdot b) \tag{13}$$

The subtraction can be implemented by inhibitory neurons or by temporal derivative in a working memory system.

Short-term Synaptic Plasticity

Another potential mechanism to generate quadratic binding is short-term synaptic plasticity. This can be accomplished by a short-term increase in residual presynaptic calcium levels in working memory enabling b to modulate the synapses G^k so that subsequent input a^T will generate the postsynaptic potential

$$\begin{aligned}\omega(w_a^k \cdot a) + (\omega + \Delta\omega)(w_b^k \cdot b) \approx \omega(w_a^k \cdot a + w_b^k \cdot b) \\ + (w_a^k \cdot a)(w_b^k \cdot b)\end{aligned} \tag{14}$$

which contains a multiplicative component of the form in Eqn. 12. Note that Eqn. 14 contain a linear term weighted by ω. This term may not need to completely cancel as it provides the trace with some similarity to both a and b, potentially allowing objects or context to be independently used as a retrieval cue. However, if ω relatively small, the trace will remain most similar to the bound conjunction $g(a,b)$.

DISCUSSION

In summary, we have proposed and analyzed a model demonstrating how multiple knowledge structures, i.e., sets of relations between pairs of items, can be represented, stored, and retrieved in Hopfield type recurrent neural network models of long-term memory. Our model hypothesizes that the entire set of relations is encoded through binding operations, summation and binarization, in a single pattern of activity in a neuronal popula-

tion, which is then stored as fixed point in the recurrent network. Retrieval of relational information from long-term memory, consists in our model of two stages: first, retrieval of the desired fixed point, and subsequent unbinding to uncover individual relations with the aid of a separate memory system, the Dictionary.

We also show that this scheme can be used to model long-term memory of temporal sequences by creating structure vectors for sequences of temporally associated items and store in a recurrent network compressed versions of the sequences as fixed points. Sequence recall consists of retrieval of the "sequence" fixed point, and subsequent unfolding of the stored events through a sequence of unbinding operations. In this application we have also demonstrated that our model for storing structure vectors in long-term memory is not very sensitive to the presence of a partial overlap between different structures.

Our analysis suggests that the success of this long-term memory system depends not only on the memory capacity of the attractor network but also very crucially on the properties of the memory basins of attraction and the landscape in the surrounding "terrain", such as the degree of overlap between "spurious" states outside the basins with the target fixed point (inside it). For this reason, a learning rule that decorrelates memories and yields smoother basins is clearly superior, as shown in Supplementary Fig. 8. Due to the dense distributed nature of the binding scheme employed here (HRR), we have not studied the effect of pattern sparsity on the long-term memory system [34]. It would be interesting to explore the sparsity effect in sparse binding schemes [49–52] and generally how the binding matrices can be learned in a biologically plausible way.

We close by briefly discussing two important aspects of this work which have the most immediate phenomenological implications. A key aspect of our model is the existence of neuronal populations representing entire relational structures in long-term memory as persistent patterns of activity displaying the "holistic context" of each structure. This system interacts with a working memory system which executes the dynamics of retrieving details of the stored relations. We have not addressed the interesting question of the mechanism by which a stream of experiences is segmented into a sequence of discrete events [53], or more generally, the mechanism that segments complex environments into a discrete sets of bound items. In particular, our model of long-term memory of sequences predicts that the retrieval of a temporal sequence is associated with a persistent pattern of activity (representing the context of the entire sequence) in addition to sequential dynamics involving the dynamic interaction between working and long-term memory. This can be tested in recordings of neuronal activity during recall of sequences in the hippocampus and in songbirds.

Finally, as mentioned above, our framework of storage and retrieval of relational knowledge structures in long-term memory relies on the existence of a complementary long-term memory system, the "Dictionary", which stores the individual building blocks comprising the relational knowledge. It is tempting to identify these two complementary memory systems as representing episodic memory (the relational system) and semantic memory (the Dictionary) [54], although we emphasize that in the present context, semantic memory does not necessarily require language and presumably exists in other species as well. The synergy of these two "complementary" memory systems results in an associative memory system with both the capacity and flexibility to store and faithfully represent complex knowledge structures in long-term memory. Adapting this framework to further explain empirically observed phenomena in memory will require adherence to known biological properties of hippocampal representations as well more explicit models of both the Dictionary and the working memory system in which binding and unbinding occurs.

METHODS

Holographic Reduced Representation

HRR [8, 55] is a commonly used VSA scheme with fixed forms for the binding and unbinding matrices in Eqns. 2 and 4. The binding operation g is given by the circular convolution operation of the vectors a and b where

$$g_k = \sum_{j=0}^{N-1} a_j b_{k-j} \ , \ k = 0, \ldots, N-1 \tag{15}$$

and all subscripts are defined modulo N. The circular convolution operation is both associative and commutative. The corresponding decoding operation ϕ is realized through circular correlation of the two vectors $\widehat{S}$ and b where

$$\hat{a}_{k\ell} = \sum_{j=0}^{N-1} \widehat{S}_j b_{j+k,\ell} \tag{16}$$

We see from comparing Eqns. 2 and 4 with Eqns. 15 and 16 that HRR corresponds to the following choice for the encoding and decoding matrices

$$G^k_{ij} = \delta_{k,j+i} \tag{17}$$

$$F^k_{ij} = \delta_{k,j-i} \tag{18}$$

The commutativity of the encoding operation implies that HRR encoding is commutative. To represent non-commutative asymmetric relations, we can simply exchange the binding and unbinding operations i.e., binding with circular correlation and unbinding with circular convolution [41]. The full details of the statistics of decoding for HRR is given in Supplementary Appendix I B.

Unbinding Accuracy

We access the typical decoding performance by considering the case in which a's and b's are random vectors with components drawn iid from $\mathcal{N}(0, \frac{1}{N})$ and the dictionaries for a's contains D elements. Then the ML decoding error is well approximated by

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \sqrt{\text{SNR}}\right)^D\right) \tag{19}$$

where $Dz = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ and $H(z) = \frac{1}{2}\text{erfc}\left(\frac{z}{\sqrt{2}}\right)$. Here SNR is a signal-to-noise-ratio defined in terms of the mean overlap of the estimator $\hat{a}_d$ with the correct Dictionary item a_d and the variance of the overlap with incorrect Dictionary item $a_{d'}$, i.e.,

$$\text{SNR} = \frac{\langle \hat{a}_d \cdot a_d \rangle^2}{\langle (\hat{a}_d \cdot a_{d'})^2 \rangle} \tag{20}$$

where $d' \neq d$, and the average is over the Gaussian distributions of the components of a_d and $a_{d'}$. For full details see Supplementary Appendix II. For SNR $\gg 1$, the decoding error can be approximated as

$$P_\epsilon \approx \sqrt{\frac{2}{\pi \text{SNR}}} D e^{-\frac{\text{SNR}}{4}} \tag{21}$$

To leading order, the SNR for many VSA binding schemes (including HRR) is

$$\text{SNR} \sim \frac{N}{L} \tag{22}$$

Eqn. 21 implies that $P_\epsilon \ll 1$ as long as $N \gtrsim O(L \log D)$. Hence, for $L \ll N$ accurate decoding requires the size of the Dictionary D be at most polynomial in N. In this regime, assumed throughout, the size of the Dictionary has little effect on performance, which is dominated by the SNR. P_ϵ and the inverse SNR are shown as functions of L in Fig. 1b and c, respectively.

Memory Network

Throughout this work, we consider Hopfield type recurrent neural networks with binary neurons. The state of the network at time t, $\sigma(t)$ is given by the update rule

$$\sigma_i(t) = \text{sgn}\left(\sum_j J_{ij}\sigma_j(t-1)\right) \tag{23}$$

where updates are done either in series or in parallel. For simplicity, parallel updates are used for the figures in the main text, but we show in Supplementary Appendix III C that the results are qualitatively similar for serial updates.

Given a set of memories σ^μ, the synaptic weight matrix J_{ij} must be chosen so that each of the memories is a fixed point of the dynamics in Eqn. 23. There are several different learning rules which can accomplish this. Mainly, we consider the Pseudo-inverse rule [36] with synaptic weight matrix given by

$$J_{ij} = \frac{1}{N}\sum_{\mu,\nu=1}^{P} \sigma_i^\mu C^{-1}_{\mu\nu}\sigma_j^\nu, \; J_{ii} = 0 \tag{24}$$

where the pattern overlap matrix $C_{\mu\nu}$ is defined as

$$C_{\mu\nu} = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^\mu \sigma_i^\nu \tag{25}$$

We also consider the Hebb rule given by

$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{P} \sigma_i^\mu \sigma_j^\mu, \; J_{ii} = 0 \tag{26}$$

and the Storkey rule [37, 56, 57] with J_{ij} given by

$$\begin{aligned} J_{ij}^\mu &= \frac{N+1}{N-1} J_{ij}^{\mu-1} + \frac{1}{N-1}(\sigma_i^\mu \sigma_j^\mu - \delta_{ij}) \\ &- \frac{1}{N-1}(\sigma_i^\mu h_j^\mu + \sigma_j^\mu h_i^\mu) \\ J_{ii}^\mu &= 0 \end{aligned} \tag{27}$$

where,

$$h_i^\mu = \sum_{k=1}^{N} J_{ik}^{\mu-1}\sigma_k^\mu \tag{28}$$

These learning rules differ in their capacity and the average size of the basins of attraction for memories at a given memory load α, further discussed Supplementary Appendix III D.

Simulations

We simulate the memory storage, retrieval, and decoding processes by creating dictionaries of objects and attributes $\mathcal{D}^a$ and $\mathcal{D}^b$ where a's and b' s are random vectors with components drawn iid from a Gaussian distribution i.e. $a_i \sim \mathcal{N}(0, \frac{1}{N})$, $b_i \sim \mathcal{N}(0, \frac{1}{N})$. The size of these dictionaries is fixed to $D = L_{max}N$, where L_{max} is the size of the largest structure being considered. In Fig. 1b we show the decoding error for several values of L_{max} and for Figs. 1c and 3 we set $L_{max} = 30$. We then use a subset of the dictionaries to create P knowledge structures with vector representations given by HRR encoding. These structures are then point-wise binarized and used to compute the synaptic weight matrix using the Pseudo-inverse rule unless otherwise stated. Since the encoding of the structures induces a similarity with individual relations

$g(a,b)$ rather than with a or b individually, we find that the same set of attributes $\{b_1, b_2, \dots b_L\}$ can be the same across several or all of the different knowledge structures while retaining the ability to decode the corresponding object a_ℓ^μ from retrieved structure σ_r^μ. Hence, we consider the case in which the same attributes are used in each structure i.e. $b_\ell^\mu = b_\ell$.

We test the performance of the memory network by initializing the network in the state $\sigma^\mu(0) = \sigma_0^\mu = \mathrm{sgn}(\widehat{S}_0^\mu)$ for each memory $\mu = 1, \dots, P$ where $S_0^\mu = \{(a_\ell^\mu, b_\ell^\mu)\}_{\ell=1}^{L_0}$ is the subset of L_0 relations used to create a retrieval cue. We then evolve the network for T parallel updates, denoting the attractor reached by the network as $\sigma_r^\mu = \sigma^\mu(T)$, i.e., the retrieved state starting from partial cue of the μ-th structure. We define m^μ as the overlap between σ^μ and σ_r^μ i.e.

$$m^\mu = \frac{1}{N} \sum_{i=1}^{N} \sigma_i^\mu \sigma_{ri}^\mu \tag{29}$$

For each retrieved structure σ_r^μ, we use b_L^μ, corresponding to a relation *not* contained in the initializing structure, to obtain an estimate $\hat{a}_L^\mu$ for a_L^μ, which then identified with the Dictionary element with which it has the highest overlap.

The Pseudo-inverse rule ensures that the basins of attraction for different structures are essentially identical regardless of potential differences in the overlap between different structures. In simulations, this allows us to consider each structure as an independent trial. The fraction of trials in which a_L^μ is incorrectly decoded from σ_r^μ provides an empirical estimate of the decoding error P_ϵ. We also construct an empirical SNR from Eqn. 20. Finally, we measure m^μ for each structure (Eqn. 29) to obtain an empirical distribution $p(m)$ where the overlaps m are calculated for each memory in a trial and accumulated over many trials. The distribution $p(m)$ does not appear to change if measured over multiple trials with different patterns or for multiple patterns within the same trial, which further supports the ability to consider each structure as a separate trial. The distribution $p(m)$ is a statistical measure of the retrieval quality for structures of fixed size L, memory load α, and retrieval cue length L_0.

Determination of l_c

To determine l_c as a function of the various network parameters, we calculate the relation between L_0/L and the average initial overlap m_0 with the desired structure in the limit of large N, yielding

$$m_0 \approx \frac{2}{\pi} \arctan\left(\left(\frac{\frac{L_0}{L}}{1-\frac{L_0}{L}}\right)^{\frac{1}{2}}\right) \tag{30}$$

Further details of the derivation are provided in Supplementary Appendix I D. Using Eqn. 30, we write $l_c(\alpha)$ in terms of $m_{min}(\alpha)$ defined in the main text as

$$l_c(\alpha) = \min \frac{L_0}{L} \approx \frac{\tan^2(\frac{\pi}{2} m_{min}^2(\alpha))}{1+\tan^2(\frac{\pi}{2} m_{min}^2(\alpha))} \tag{31}$$

To determine $m_{min}(\alpha)$ we resort to the Pseudo-inverse model with random binary patterns as memories [36], which is simpler to simulate. Results relating $m_{min}(\alpha)$ and l_c are shown in Supplementary Fig. 1.

Empirical Distribution of Overlaps

We find that the empirical distribution $p(m)$ is bimodal and takes the general form

$$p(m) = (1-p_1)p_{m<1}(m) + p_1\delta(m-1) \tag{32}$$

where p_1 is the probability that a structure is perfectly retrieved from memory and $p_{m<1}(m)$ corresponds to the distribution of m for imperfectly retrieved memories.

The peak at $m = 1$ corresponds to trajectories converging to the target memories. This can be nonzero even when initial overlap m_0 is outside the mean basin radius, indicating non-spherical basin shape. The second mode, peaked at $0 < m < 1$ results from trajectories that converged to a fixed point outside the basin with a significant residual overlap with the memory. We characterize the shape of the distribution by the probability of $m = 1$, p_1, the width of the lower m mode, σ_m and the mean of that mode, m^*. Results are shown in Supplementary Fig. 4a, for several values of N and two values of α.

As noted in [36, 38], the shape of the distribution $p(m)$ is sensitive to finite size effects. To analyze these effects, we calculate $p(m)$ for different sizes in a standard Pseudo-inverse model where the initial overlap m_0 can essentially be varied continuously. For $m_0 > m_{min}(\alpha)$ almost all trials converge to the memorized pattern. For a range of values $m_0 < m_{min}(\alpha)$, $p(m)$ is bimodal. We find that $p(m)$ obtained from networks storing random patterns is very similar to the distribution obtain from networks storing structure memories, when the m_0 and L_0/L are related as in Eqn. 30. We find that for large N, p_1 approaches a step function changing from zero to one as m_0 crosses $m_{min}(\alpha) = 1 - R(\alpha)$. Near this transition, p_1 can be approximated as

$$p_1 \sim \frac{1}{1+e^{-\sqrt{N}(m_0 - m_{min}(\alpha))}} \tag{33}$$

indicating that it converges to a step function exponentially fast with $\sqrt{N}$. In addition, σ_m is very small outside the narrow transition regime of m_0 and shrinks to zero everywhere as $1/\sqrt{N}$. From this, we conclude that for $N \to \infty$, $p(m)$ becomes a δ function, which is either located at $m = 1$ for $m_0 > m_{min}$ or at a smaller value m^* which increases smoothly with m_0, starting from zero and

reaching 1 as m_0 increases from zero to m_{min}. Thus in large networks, the basins have a roughly spherical shape, such that virtually all initial conditions with $m_0 \geq m_{min}$ converge to the memory, and all initial conditions with $m_0 < m_{min}$ converge to fixed points with partial overlap, m^*.

Acknowledgements

We thank Tankut Can, Naoki Hiratani, Mikhail Katkov, Kamesh Krishnamurthy, Andrew Saxe, Nimrod Shaham, and Misha Tsodyks for fruitful discussions. J.S. acknowledges support from the NSF through the Center for the Physics of Biological Function (PHY-1734030) and computational resources from the Princeton Research Computing at Princeton University, a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing. H.S. acknowledges support from the Swartz Program in Theoretical Neuroscience at Harvard, the Gatsby Charitable Foundation, and NIH grant NINDS (1U19NS104653).

[1] E. Tulving, Episodic and semantic memory, in *Organization of memory*, edited by E. Tulving and W. Donaldson (Academic Press, 1972) pp. 381–403.

[2] M. W. Howard and M. J. Kahana, A distributed representation of temporal context, Journal of Mathematical Psychology **46**, 269 (2002).

[3] E. C. Tolman, Cognitive maps in rats and men., Psychological Review **55**, 189 (1948).

[4] J. R. Manns and H. Eichenbaum, A cognitive map for object memory in the hippocampus, Learning and Memory **16**, 616 (2009).

[5] T. E. Behrens, T. H. Muller, J. C. Whittington, S. Mark, A. B. Baram, K. L. Stachenfeld, and Z. Kurth-Nelson, What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior, Neuron **100**, 490 (2018).

[6] P. Smolensky, Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial Intelligence **46**, 159 (1990).

[7] T. A. Plate, Distributed representations and nested compositional structure, (1994).

[8] C. Eliasmith, *How to Build a Brain: A Neural Architecture for Biological Cognition* (Oxford University Press, 2013).

[9] J. Feldman and D. Ballard, Connectionist models and their properties, Cognitive Science **6**, 205 (1982).

[10] K. J. Holyoak and J. E. Hummel, The proper treatment of symbols in a connectionist architecture., in *Cognitive dynamics: Conceptual and representational change in humans and machines.* (Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US, 2000) pp. 229–263.

[11] P. Smolensky, On the proper treatment of connectionism, in *Philosophy, Mind, and Cognitive Inquiry: Resources for Understanding Mental Processes*, edited by D. J. Cole, J. H. Fetzer, and T. L. Rankin (Springer Netherlands, Dordrecht, 1990) pp. 145–206.

[12] J. Feldman, The neural binding problem(s), Cognitive Neurodynamics **7**, 1 (2013).

[13] R. Jackendorff, *Foundations of language: Brain, meaning, grammar, evolution* (Oxford University Press, 2002).

[14] K. Greff, S. van Steenkiste, and J. Schmidhuber, On the Binding Problem in Artificial Neural Networks, (2020), arXiv:2012.05208.

[15] O. Barak, M. Rigotti, and S. Fusi, The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off, Journal of Neuroscience **33**, 3844 (2013).

[16] W. F. Podlaski, E. J. Agnes, and T. P. Vogels, Context-modular memory networks support high-capacity, flexible, and robust associative memories, bioRxiv , 2020.01.08.898528 (2020).

[17] P. Kanerva, Binary spatter-coding of ordered k-tuples, in *Artificial Neural Networks - ICANN 96, 1996 International Conference, Bochum, Germany, July 16-19, 1996, Proceedings* (1996) pp. 869–873.

[18] S. D. Levy and R. Gayler, Vector symbolic architectures: A new building material for artificial general intelligence, in *Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference* (IOS Press, NLD, 2008) p. 414–418.

[19] D. A. Rachkovskij and E. M. Kussul, Binding and normalization of binary sparse distributed representations by context-dependent thinning, Neural Computation **13**, 411 (2001), https://doi.org/10.1162/089976601300014592.

[20] D. Kleyko, D. A. Rachkovskij, E. Osipov, and A. Rahimi, A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges, , 1 (2021), arXiv:2112.15424.

[21] T. C. Stewart, T. Bekolay, and C. Eliasmith, Neural representations of compositional structures: representing and manipulating vector spaces with spiking neurons, Connection Science **23**, 145 (2011).

[22] K. Schlegel, P. Neubert, and P. Protzel, *Artificial Intelligence Review*, 0123456789 (Springer Netherlands, 2021) arXiv:2001.11797.

[23] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, Relational inductive biases, deep learning, and graph networks (2018), arXiv:1806.01261 [cs.LG].

[24] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, A simple neural network module for relational reasoning, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).

[25] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, Deep sets, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017) pp. 3391–3401.

[26] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference **1**, 4171 (2019), arXiv:1810.04805.

[27] E. P. Frady, D. Kleyko, and F. T. Sommer, A Theory of Sequence Indexing and Working Memory in Recurrent Neural Networks, Neural Computation **30**, 1449 (2018).

[28] J. C. Whittington, T. H. Muller, S. Mark, G. Chen, C. Barry, N. Burgess, and T. E. Behrens, The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation, Cell **183**, 1249 (2020).

[29] J. C. R. Whittington, J. Warren, and T. E. J. Behrens, Relating transformers to models and neural representations of the hippocampal formation, (2021), arXiv:2112.04035.

[30] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Storing infinite numbers of patterns in a spin-glass model of neural networks, Phys. Rev. Lett. **55**, 1530 (1985).

[31] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences **79**, 2554 (1982), https://www.pnas.org/content/79/8/2554.full.pdf.

[32] H. Sompolinsky and I. Kanter, Temporal association in asymmetric neural networks, Phys. Rev. Lett. **57**, 2861 (1986).

[33] X. Xie, R. Hahnloser, and H. S. Seung, Groups of Neurons in Lateral Inhibitory Networks, .

[34] M. V. Tsodyks and M. V. Feigel'man, The enhanced storage capacity in neural networks with low activity level, Europhysics Letters (EPL) **6**, 101 (1988).

[35] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Spin-glass models of neural networks, Phys. Rev. A **32**, 1007 (1985).

[36] I. Kanter and H. Sompolinsky, Associative recall of memory without errors, Phys. Rev. A **35**, 380 (1987).

[37] A. Storkey and R. Valabregue, The basins of attraction of a new hopfield learning rule, Neural Networks **12**, 869 (1999).

[38] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Statistical mechanics of neural networks near saturation, Annals of Physics **173**, 30 (1987).

[39] G. D. Logan, Automatic control: How experts act without thinking, Psychological Review **125**, 453 (2018).

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

[41] M. Nickel, L. Rosasco, and T. Poggio, Holographic embeddings of knowledge graphs, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16 (AAAI Press, 2016) p. 1955–1961.

[42] R. A. Silver, Neuronal arithmetic, Nature Reviews Neuroscience **11**, 474 (2010).

[43] W. H. Mehaffey, B. Doiron, L. Maler, and R. W. Turner, Deterministic multiplicative gain control with active dendrites, Journal of Neuroscience **25**, 9968 (2005).

[44] Y. Ahmadian, D. B. Rubin, and K. D. Miller, Analysis of the Stabilized Supralinear Network, Neural Computation **25**, 1994 (2013).

[45] P. Y. Deng and V. A. Klyachko, The diverse functions of short-term plasticity components in synaptic computations, Communicative and Integrative Biology **4**, 543 (2011).

[46] R. S. Zucker and W. G. Regehr, Short-term synaptic plasticity, Annual Review of Physiology **64**, 355 (2002).

[47] J. Ba, G. E. Hinton, V. Mnih, G. Deepmind, J. Z. Leibo, and C. Ionescu, [2016-NIPS] Using Fast Weights to Attend to the Recent Past, , 1 (2016).

[48] I. Schlag, K. Irie, and J. Schmidhuber, Linear Transformers Are Secretly Fast Weight Programmers, (2021), arXiv:2102.11174.

[49] E. P. Frady, D. Kleyko, and F. T. Sommer, Variable Binding for Sparse Distributed Representations: Theory and Applications, IEEE Transactions on Neural Networks and Learning Systems , 1 (2021).

[50] D. A. Rachkovskij, Representation and processing of structures with binary sparse distributed codes, IEEE Transactions on Knowledge and Data Engineering **13**, 261 (2001).

[51] D. A. Rachkovskij, E. M. Kussul, and T. N. Baidyk, Building a world model with structure-sensitive sparse binary distributed representations, Biologically Inspired Cognitive Architectures **3**, 64 (2013).

[52] N. Hiratani and H. Sompolinsky, Optimal quadratic binding for relational reasoning in vector symbolic neural architectures, , 1 (2022), arXiv:2204.07186.

[53] N. T. Franklin, K. A. Norman, C. Ranganath, J. M. Zacks, and S. J. Gershman, Structured Event Memory: A neuro-symbolic model of event cognition., Psychological Review **127**, 327 (2020).

[54] A. A. Kumar, Semantic memory: A review of methods, models, and current challenges, Psychonomic Bulletin Review **28**, 40 (2021).

[55] T. A. Plate, Holographic reduced representations, IEEE Transactions on Neural Networks **6**, 623 (1995).

[56] A. Storkey, Increasing the capacity of a hopfield network without sacrificing functionality, in *Proceedings of the 7th International Conference on Artificial Neural Networks*, ICANN '97 (Springer-Verlag, Berlin, Heidelberg, 1997) p. 451–456.

[57] A. Storkey and R. Valabregue, A hopfield learning rule with high capacity storage of time-correlated patterns (1997).

Supplementary Appendix: Associative memory of structured knowledge

Julia Steinberg[1,2,3,*] and Haim Sompolinsky[2,4,†]

[1] *Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA*
[2] *Center for Brain Science, Harvard University, Cambridge MA 02138, USA*
[3] *Joseph Henry Laboratories of Physics, Princeton University, Princeton NJ 08544, USA*
[4] *Edmond and Lily Safra Center for Brain Sciences, Hebrew University, Jerusalem 91904, Israel*

CONTENTS

I. Holographic Reduced Representation 1
A. Overview 1
B. Clean-up and SNR 2
C. Binarization and Empirical SNR 4
D. Memory Initialization 5

II. Decoding Error 6
A. Limitations for Good Decoding 7
B. Decoding After Memory Retrieval 8

III. Insights from Random Patterns 9
A. Network Order Parameters 9
B. Overlap Distributions 10
C. Retrieval Dynamics 10
D. Comparing Learning Rules 12
E. Connection to Storage of Knowledge Structures 16

IV. Temporal Sequences as Sequences of Attractors 16

References 17

I. HOLOGRAPHIC REDUCED REPRESENTATION

A. Overview

In holographic reduced representation (HRR) [1], the binding operation g is circular convolution

$$g = \mathrm{cconv}(a, b) \tag{1}$$

where the individual components of g are given by

$$g_j = \sum_{k=0}^{N-1} a_k b_{j-k} = \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} \delta_{j,k+m} a_k b_m \ , \ j = 0, \ldots, N-1 \tag{2}$$

with all subscripts are defined modulo N. For objects and attributes $a, b \in \mathbb{R}^N$, this creates a representation $g(a, b) \in \mathbb{R}^N$. The circular convolution operation is both associative and commutative, i.e. $\mathrm{cconv}(a, b) = \mathrm{cconv}(b, a)$.

The corresponding decoding operation is given by the circular correlation

$$f = \mathrm{ccorr}(\widehat{S}, b) \tag{3}$$

* jsteinberg@princeton.edu
† haim@fiz.huji.ac.il

where the individual components of f are given by

$$f_j = \sum_{k=0}^{N-1} \widehat{S}_k b_{j+k,\ell} = \sum_{k=0}^{N-1}\sum_{m=0}^{N-1} \delta_{j,m-k}\widehat{S}_k b_m \ , \ j = 0, \ldots, N-1 \tag{4}$$

In general the circular correlation operation is neither associative nor commutative. However, it can be related to circular convolution by defining the involution of a vector a as a^* where $a^*_j = a_{-j}$. Then $\mathrm{ccorr}(a, b) = \mathrm{cconv}(a^*, b)$ and a^* as serves as an approximate inverse to a when both a and b are random vectors.

B. Clean-up and SNR

To quantify the effect of the ML clean-up operation described in the main text, we compute various statistics of the estimator $\hat{a}_\ell$ for the corresponding object a_ℓ decoded from a relational structure $\widehat{S}$ using the HRR unbinding operation. We consider the case where all objects and attributes are random vectors with components drawn iid, i.e., $a_{id} \sim \mathcal{N}(0, \frac{1}{N})$, $b_{id} \sim \mathcal{N}(0, \frac{1}{N})$ and decoding is done from a dictionary of size D.

Since the encoding operation is permutation invariant, the statistics for all $\hat{a}_\ell$ decoded from the full structure should be the same. For simplicity we set $\ell = 1$ and consider $\hat{a}_1$. We index dictionary items so that indices $\ell < L$ correspond to the dictionary items contained in the structure $\widehat{S}$ and indices $L < \ell < D$ corresponds to dictionary items that are not contained in $\widehat{S}$. Note that

$$\langle a_{i,\ell} a_{j,\ell'} \rangle = \frac{1}{N}\delta_{ij}\delta_{\ell\ell'} \tag{5}$$

$$\langle a_{i,\ell} a_{j,\ell'} a_{k,s} a_{l,s'} \rangle = \frac{1}{N^2}(\delta_{ij}\delta_{\ell\ell'}\delta_{kl}\delta_{ss'} + \delta_{ik}\delta_{\ell s}\delta_{jl}\delta_{\ell' s'}\delta_{ss'} + \delta_{il}\delta_{\ell s'}\delta_{jk}\delta_{\ell' s}) \tag{6}$$

Using the expression for the unbinding operation in Eqn. 4, we can express the estimator $\hat{a}_1$ as

$$\hat{a}_{j,1} = \sum_{k=0}^{N-1} \widehat{S}_k b_{j+k,1} \tag{7}$$

Using Eqn. 2, we can express this in terms of a_ℓ's and a_ℓ's, and $\hat{a}_{j,1}$ as

$$\begin{aligned} \hat{a}_{j,1} &= a_{j,1}\sum_{k=0}^{N-1}(b_{k,1})^2 + \sum_{k=0,k'\neq k}^{N-1} a_{k-k'+j,1} b_{k,1} b_{k',1} + \sum_{\ell=2}^{L}\sum_{k=0}^{N-1}\sum_{k'=0}^{N-1} a_{k-k'+j,\ell} b_{k,\ell} b_{k',1} \\ &= (1+\epsilon)\, a_{j,1} + \xi_{j,1} + \xi_{j,2} \end{aligned} \tag{8}$$

where we have defined the three noise terms as

$$\epsilon = \sum_{k=0}^{N-1}(b_{k,1})^2 - 1 \tag{9}$$

$$\xi_{j,1} = \sum_{k=0}^{N-1}\sum_{k'=0}^{N-1}(1-\delta_{kk'}) b_{k,1} b_{k',1} a_{k-k'+j,1} \tag{10}$$

$$\xi_{j,2} = \sum_{\ell=2}^{L}\sum_{k=0}^{N-1}\sum_{k'=0}^{N-1} b_{k,1} b_{k',\ell} a_{k-k'+j,\ell} \tag{11}$$

Here, ϵ is noise due to deviations in the normalization of a, $\xi_{j,1}$ is noise from interference elements of different elements of the cue within the same attribute b_1, and $\xi_{j,2}$ and $\xi_{j,2}$ are noise coming from interference between attribute b_1 with all other attributes b_ℓ contained in the structure. Since components of a_1 appear an odd number of times in each term of the sums in $\xi_{j,1}$ and $\xi_{j,2}$, we conclude that $\langle \xi_{j,1} \rangle = \langle \xi_{j,2} \rangle = 0$. Likewise, $\langle \epsilon \rangle = 0$. Using Eqns. 5 and 6, we

find that the second moments and correlation of the two noise terms are given by

$$\begin{aligned}
\langle \epsilon^2 \rangle &= 1 + \left(\sum_{k=0}^{N-1} \sum_{q=0}^{N-1} \langle b_k^2 b_q^2 \rangle \right) - 2 \sum_{k=0}^{N-1} \langle b_k^2 \rangle \\
&= \sum_{k=0}^{N-1} \sum_{q=0}^{N-1} \frac{1}{N^2} (1 + 2\delta_{kq}) - 1 \\
&= \frac{2}{N}
\end{aligned} \tag{12}$$

$$\begin{aligned}
\langle \xi_{j,1}^2 \rangle &= \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} (1 - \delta_{kk'})(1 - \delta_{qq'}) \langle b_{k,1} b_{k',1} b_{q,1} b_{q',1} \rangle \langle a_{k-k'+j,1} a_{q-q'+j,1} \rangle \\
&= \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \frac{1}{N^3} (\delta_{kq} \delta_{k'q'} - \delta_{kk'} \delta_{qq'} \delta_{kq}) \\
&= \frac{(N-1)}{N^2}
\end{aligned} \tag{13}$$

$$\begin{aligned}
\langle \xi_{j,2}^2 \rangle &= \sum_{\ell=2}^{L} \sum_{\ell'=2}^{L} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \langle b_{k,1} b_{k',\ell} b_{q,1} b_{q',\ell'} \rangle \langle a_{k-k'+j,\ell} a_{q-q'+j,\ell'} \rangle \\
&= \sum_{\ell=2}^{L} \sum_{\ell'=2}^{L} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \frac{1}{N^3} (\delta_{1\ell} \delta_{\ell\ell'} \delta_{kk'} \delta_{qq'} + \delta_{\ell\ell'} \delta_{kq} \delta_{k'q'} + \delta_{1\ell} \delta_{\ell\ell'} \delta_{kq'} \delta_{k'q} \delta_{k-k',q-q'}) \\
&= \frac{L}{N} + \frac{1}{N^2}
\end{aligned} \tag{14}$$

$$\begin{aligned}
\langle \xi_{j,1} \xi_{j,2} \rangle &= \sum_{\ell=2}^{L} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} (1 - \delta_{kk'}) \langle b_{k,1} b_{k',1} b_{q,1} b_{q',\ell} \rangle \langle a_{k-k'+j,1} a_{q-q'+j,\ell} \rangle \\
&= 0
\end{aligned} \tag{15}$$

and from Eqn. 8, we see the noise in each component of the estimator is

$$\frac{\langle \delta \hat{a}_{j,\ell} \rangle}{\langle a_{j,\ell}^2 \rangle} \sim L \tag{16}$$

We now calculate the first two moments of the overlap of the estimator with the correct pattern $a_d \cdot \hat{a}_1$ to obtain the SNR for HRR defined in Eqn. 20 of Methods. $a_d \cdot \hat{a}_1$ can be expressed as

$$a_d \cdot \hat{a}_1 = (1 + \epsilon)\, a_d \cdot a_1 + a_d \cdot (\xi_1 + \xi_2) \tag{17}$$

We see from Eqn. 17 that $\langle a_1 \cdot \hat{a}_1 \rangle = 1$ and $\langle a_d \cdot \hat{a}_1 \rangle = 0$, so the estimator is unbiased. The second moment is given by

$$\begin{aligned}
\langle (a_d \cdot \hat{a}_1)^2 \rangle &= \Big\langle \sum_{j=0}^{N-1} \sum_{j'=0}^{N-1} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \sum_{\ell=1}^{L} \sum_{\ell'=1}^{L} b_{k,1} b_{k',\ell} a_{k-k'+j,\ell} a_{j,d} b_{q,1} b_{q',\ell'} a_{q-q'+j',\ell'} a_{j',d} \Big\rangle \\
&= \sum_{j=0}^{N-1} \sum_{j'=0}^{N-1} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \sum_{\ell=1}^{L} \sum_{\ell'=1}^{L} \langle b_{k,1} b_{k',\ell} b_{q,1} b_{q',\ell'} \rangle \langle a_{k-k'+j,\ell} a_{j,d} a_{q-q'+j',\ell'} a_{j',d} \rangle \\
&= \sum_{j=0}^{N-1} \sum_{j'=0}^{N-1} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{q=0}^{N-1} \sum_{q'=0}^{N-1} \sum_{\ell=1}^{L} \sum_{\ell'=1}^{L} \frac{1}{N^4} (\delta_{kk'} \delta_{qq'} \delta_{d\ell} \delta_{1\ell'} \delta_{\ell\ell'} + \delta_{kk'} \delta_{qq'} \delta_{kq} \delta_{\ell\ell'} \delta_{1\ell} + \delta_{kk'} \delta_{qq'} \delta_{jj'} \delta_{d\ell} \delta_{1\ell'} \delta_{\ell\ell'} \\
&+ \delta_{kk'} \delta_{qq'} \delta_{kq} \delta_{\ell\ell'} \delta_{\ell d} + \delta_{kq} \delta_{k'q'} \delta_{jj'} \delta_{\ell\ell'} + \delta_{kq} \delta_{k'q'} \delta_{jj'} \delta_{\ell\ell'} \delta_{\ell d} + \delta_{kk'} \delta_{qq'} \delta_{kq} \delta_{1\ell'} \delta_{d\ell} \delta_{\ell\ell'} + \delta_{kk'} \delta_{qq'} \delta_{kq} \delta_{jj'} \delta_{\ell\ell'} \delta_{1\ell} \\
&+ \delta_{kk'} \delta_{qq'} \delta_{kq} \delta_{jj'} \delta_{1\ell} \delta_{d\ell'} \delta_{\ell\ell'}) \\
&= \delta_{1d} + \frac{L}{N} + \frac{1}{N} (1 + 2\delta_{1d} + 2\Theta(L-d)) + \frac{1}{N^2} (1 + \delta_{1d}))
\end{aligned} \tag{18}$$

To summarize

$$\langle a_d \cdot \hat{a}_1 \rangle = \begin{cases} 1 & d = 1 \\ 0 & 1 < d \leq L \\ 0 & L < d \end{cases} \tag{19}$$

$$\Sigma_{1d}^2 = \langle (a_d \cdot \hat{a}_1 - \langle a_d \cdot \hat{a}_1 \rangle)^2 \rangle = \begin{cases} \frac{L+5}{N} + \frac{2}{N^2} & d = 1 \\ \frac{L+3}{N} + \frac{1}{N^2} & 1 < d \\ \frac{L+1}{N} + \frac{1}{N^2} & L < d \end{cases} \tag{20}$$

For $L \ll N$, the SNR for overlaps defined in Eqn. 20 of the main text is then approximately

$$\frac{\langle a_1 \cdot \hat{a}_1 \rangle^2}{\langle (a_d \cdot \hat{a}_1)^2 \rangle} \approx \frac{N}{L} \tag{21}$$

C. Binarization and Empirical SNR

When decoding from a binarized structure $\sigma = \text{sgn}(\widehat{S})$, the estimator $\hat{a}_1$ is given by

$$\hat{a}_{j1} = \sum_{k=0}^{N-1} \sigma_k b_{k+j,1} \tag{22}$$

where

$$\sigma_j = \text{sgn}\left(\sum_{k=0}^{N-1} \sum_{\ell=1}^{L} a_{k,\ell} b_{j-k,\ell} \right) \tag{23}$$

To compute the SNR, we start by evaluating the overlap between the estimator $\hat{a}_1$ with items in the dictionary $\langle a_d \cdot \hat{a}_1 \rangle$ which can be expressed as

$$\langle a_d \cdot \hat{a}_1 \rangle = \Big\langle \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} a_{j,d} b_{k,1} \text{sgn}\left(a_{j,1} b_{k,1} + \sum_{q \neq k}^{N-1} a_{k-q+j,1} b_{q,1} + \sum_{\ell=2}^{L} \sum_{q=0}^{N-1} a_{k-q+j,\ell} b_{q,\ell} \right) \Big\rangle \tag{24}$$

While individual products of elements $a_{\ell,i} b_{\ell,j}$ are not Gaussian distributed, the expectation $\langle a_d \cdot \hat{a}_1 \rangle$ is the sum of N^2 random variables with correlations only occurring at higher order as the expectation of the product of different elements of each circular convolution is zero. Since all terms contained in the sums over indiced j and k in Eqn. 24 are only correlated at higher order, we can approximate each term in the sum as independent which gives us

$$\langle a_d \cdot \hat{a}_1 \rangle \approx N^2 \Big\langle a_{0,d} b_{0,1} \left(\text{sgn}\left(a_{0,1} b_{0,1} + \sum_{q=1}^{N-1} a_{N-q,1} b_{q,1} + \sum_{\ell=2}^{L} \sum_{q=0}^{N-1} a_{N-q,\ell} b_{q,\ell} \right) \right) \Big\rangle \tag{25}$$

We now define the following three random variables

$$z_1 = a_{0,1} b_{0,1} \tag{26}$$

$$z_d = a_{0,d} b_{0,1} \tag{27}$$

$$\xi = \sum_{q=1}^{N-1} a_{N-q,1} b_{q,1} + \sum_{\ell=2}^{L} \sum_{q=0}^{N-1} a_{N-q,\ell} b_{q,\ell} \tag{28}$$

where x_1 is a signal term and ξ is a noise term within the sign function. $\langle z_1 \rangle = \langle z_2 \rangle = \langle \xi \rangle = 0$ and the variances are given by

$$\Sigma_{z_1}^2 = \frac{1}{N^2} \tag{29}$$

$$\Sigma_{z_\ell}^2 = \frac{1}{N^2} \tag{30}$$

$$\Sigma_{\xi}^2 = \frac{(N-1)}{N^2} + \frac{L}{N} \sim \frac{L}{N} \tag{31}$$

For $d = 1$, $z_1 = z_\ell = z$ and $L \gg 1$ we can treat z and ξ as Gaussian. In this approximation $\langle a_1 \cdot \hat{a}_1 \rangle$ becomes

$$\begin{aligned} \langle a_1 \cdot \hat{a}_1 \rangle &\approx N^2 \int_{-\infty}^{\infty} \frac{dz}{\sqrt{2\pi\Sigma_z^2}} \int_{-\infty}^{\infty} \frac{d\xi}{\sqrt{2\pi\Sigma_\xi^2}} e^{-\frac{z^2}{2\Sigma_z^2} - \frac{\xi^2}{2\Sigma_\xi^2}} z \mathrm{sgn}\left(z + \xi\right) \\ &= \sqrt{\frac{2}{\pi}} \frac{N^2 \Sigma_z^2}{\sqrt{\Sigma_z^2 + \Sigma_\xi^2}} \\ &\sim \sqrt{\frac{2N}{\pi L}} \end{aligned} \tag{32}$$

Likewise, the second moment is given by

$$\langle (a_1 \cdot \hat{a}_1)^2 \rangle = \langle (a_d \cdot \hat{a}_1)^2 \rangle \approx N^2 \int_{-\infty}^{\infty} \frac{dz}{\sqrt{2\pi\Sigma_z^2}} e^{-\frac{z^2}{2\Sigma}} \int_{-\infty}^{\infty} \frac{d\xi}{\sqrt{2\pi\Sigma_\xi^2}} e^{-\frac{\xi^2}{2\Sigma_\xi^2}} z^2 = 1 \tag{33}$$

Putting Eqns. 32 and 33 together, the SNR of overlap when decoding a_1 from the binarized structure σ is given by

$$\mathrm{SNR} = \frac{\langle a_1 \cdot \hat{a}_1 \rangle^2}{\langle (a_d \cdot \hat{a}_1)^2 \rangle} \sim \frac{2}{\pi} \frac{N}{L} \tag{34}$$

We see from Eqn. 21 that binarizing $\widehat{S}$ has decreased the SNR by a constant factor of $\frac{2}{\pi}$.

D. Memory Initialization

We now determine the average initial overlap m_0 between a binarized retrieval structure σ_0 containing L_0 of the L relations in the binarized full structure σ in the case of random a_ℓ's and b_ℓ's are random vectors with components drawn iid as before. An expression for the average overlap m_0 between σ_0 and σ can be written as

$$m_0 = \left\langle \frac{1}{N} \sum_{j=0}^{N-1} \mathrm{sgn}\left(\widehat{S}_j\right) \mathrm{sgn}\left(\widehat{S}_{j,0}\right) \right\rangle \tag{35}$$

The overlap between components of the unbinarized structures $\widehat{S}_j$ and $\widehat{S}_{j'0}$ with HRR binding is given by

$$\begin{aligned} \langle \widehat{S}_j \widehat{S}_{j'0} \rangle &= \sum_{\ell=1}^{L} \sum_{\ell'=1}^{L_0} \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \sum_{p=0}^{N-1} \sum_{p'=0}^{N-1} \delta_{p',j'-k'} \delta_{p,j-k} a_{k,\ell} a_{k',\ell'} b_{p,\ell} b_{p',\ell'} \Big\rangle \\ &= \Big\langle \sum_{\ell=1}^{L_0} \sum_{k=0}^{N-1} \delta_{jj'} a_{k,\ell}^2 b_{j-k,\ell}^2 \Big\rangle \\ &= \frac{L_0}{N} \delta_{jj'} \end{aligned} \tag{36}$$

For large N we can approximate the sum in Eqn. 35 by treating the individual terms as as independent since there are no lower order correlations between terms. We can then define the random variables $z_1 = \widehat{S}_{j0}$ and $z_2 = \widehat{S}_j - \widehat{S}_{j0}$

as approximately Gaussian distributed with zero mean and variances $\Sigma^2_{z_1} = \frac{L_0}{N}$ and $\Sigma^2_{z_2} = \frac{(L-L_0)}{N}$ respectively. Then, Eqn. 35 can be approximated as

$$\begin{aligned} m_0 &\approx \int_{-\infty}^{\infty} \frac{dz_1}{\sqrt{\Sigma^2_{z_1}}} e^{-\frac{z_1^2}{2\Sigma^2_{z_1}}} \int_{-\infty}^{\infty} \frac{dz_2}{\sqrt{2\pi\Sigma^2_{z_2}}} e^{-\frac{z_2^2}{2\Sigma^2_{z_2}}} \mathrm{sgn}(z_1)\mathrm{sgn}(z_1+z_2) \\ &\approx \frac{2}{\pi} \arctan\left(\frac{\Sigma_{z_1}}{\Sigma_{z_2}}\right) \\ &\approx \frac{2}{\pi} \arctan\left(\left(\frac{\frac{L_0}{L}}{1-\frac{L_0}{L}}\right)^{\frac{1}{2}}\right) \end{aligned} \tag{37}$$

For $L_0 \ll L$ Eqn. 37 can be further approximated as

$$m_0 \approx \frac{2}{\pi}\sqrt{\frac{L_0}{L}} \tag{38}$$

This implies that creating retrieval structures with L_0 out of L relations will result in retrieval cues with average overlap m_0 with the full memorized structure. The variance of the distribution of initial overlaps will vanish as $N \to \infty$.

We use this relation to determine $l_c(\alpha)$ as a function of L_0/L using the value of $m_{min}(\alpha)$ obtained for a Pseudo-inverse model with random binary patterns as memories [2]. Results for $m_{min}(\alpha)$ are shown in Fig. 1a demonstrating a linear increase in the range $\alpha < 0.5$. Substituting these results in Eqn. 31 closely predicts the value of $l_c(\alpha)$ as shown in Fig. 1b.

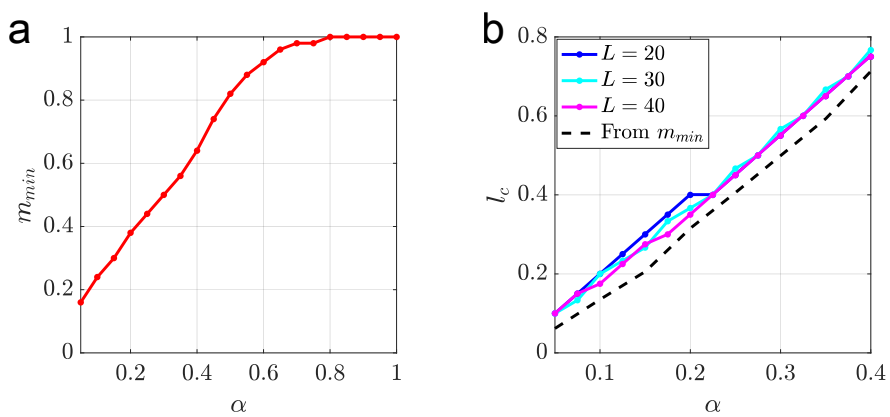

FIG. 1. (a) m_{min} shown as a function of α for a Pseudo-inverse network storing with random patterns. (b) For structured memories, m_{min} corresponds to a minimum fraction of relations needed obtain a final overlap $m = 1$, shown for $L = 20, 30, 40$ along with the value predicted by m_{min} from the expression in Eqn. 31. $N = 1000$ and $T = 20$ parallel updates are used in memory retrieval for both figures and the averages are computed over 50 trials.

II. DECODING ERROR

We derive a good approximation for the ML decoding error in terms of D, the size of the decoding dictionary, and SNR, the signal-to-noise ratio of overlaps defined in Eqn. 20 of Methods. The overlap $\mathcal{O}_{d1}$ between a dictionary item a_d with the estimator $\hat{a}_1$ for a_1 decoded from a structure $\widehat{S}$ is given by

$$\mathcal{O}_{1d} = \hat{a}_1 \cdot a_d \tag{39}$$

where as in the previous section, we index items so that $d \leq L$ corresponds to overlap with patterns within $\widehat{S}$ and $L < d$ corresponds to overlaps with other items in the Dictionary not contained in $\widehat{S}$. For ML decoding, an error

occurs if $\max(\mathcal{O}_{12},\ldots,\mathcal{O}_{1D}) \geq \mathcal{O}_{11}$. The probability of error P_ϵ is then given by

$$P_\epsilon = 1 - \int_{-\infty}^{\infty} d\mathcal{O}_{11} P(\mathcal{O}_{11}) \left(\int_{-\infty}^{\mathcal{O}_{11}} d\mathcal{O}_{1L} P(\mathcal{O}_{1L}) \right)^{L-1} \left(\int_{-\infty}^{\mathcal{O}_{11}} d\mathcal{O}_{1D} P(\mathcal{O}_{1D}) \right)^{D-L} \tag{40}$$

For large N, each $\mathcal{O}_{1d}$ essentially behaves as an independent Gaussian random variable with mean μ_{1d} and variance Σ_{1d}. For unbiased decoding schemes, $\mu_{1L} = \mu_{1D} = 0$. In general, we can consider decoding from a structure containing errors by setting $\mu_{11} = \mu$, where μ is related to the overlap of the corrupted structure with the correct structure. Then P_ϵ can be very well approximated as

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz \left(1 - H\left(-\frac{\Sigma_{11} z + \mu}{\Sigma_{1L}} \right)^{L-1} H\left(-\frac{\Sigma_{11} z + \mu}{\Sigma_{1D}} \right)^{D-L} \right) \tag{41}$$

where $Dz = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$.

For HRR $\Sigma_{11} \approx \Sigma_{1L} \approx \Sigma_{1D}$ to leading order in $\frac{1}{N}$. Defining Σ to be the leading order term in the variance, we obtain

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \frac{\mu}{\Sigma} \right)^{D} \right) \tag{42}$$

Identifying $\frac{\mu^2}{\Sigma^2}$ with the SNR of overlaps, Eqn. 42 becomes

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \sqrt{\mathrm{SNR}} \right)^{D} \right) \tag{43}$$

given in the main text.

A. Limitations for Good Decoding

We can understand the dependence of P_ϵ on the SNR and D in the large SNR regime by obtaining a saddlepoint approximation for the expression for P_ϵ given in Eqn. 43. We start by rewriting Eqn. 43 as

$$P_\epsilon \approx \int_{-\infty}^{\infty} \frac{dz}{\sqrt{2\pi}} e^{f(z)} \tag{44}$$

where $f(z)$ is given by

$$f(z) = -\frac{z^2}{2} + \log\left(1 - H\left(-z - \sqrt{\mathrm{SNR}} \right)^{D} \right) \tag{45}$$

From Eqn. 42, we see $P_\epsilon \ll 1$ for $H(z + \sqrt{\mathrm{SNR}}) \ll 1$ as $H(-z - \sqrt{\mathrm{SNR}}) = 1 - H(z + \sqrt{\mathrm{SNR}})$. So we are interested in the regime where $1 \ll z + \sqrt{\mathrm{SNR}}$. In this regime, $f(z)$ is very large, so the integral in P_ϵ can be approximated its saddlepoint value

$$P_\epsilon \approx \frac{1}{\sqrt{|f''(z_0)|}} e^{f(z_0)} \tag{46}$$

where z_0 is given by the solution of

$$z = -\frac{DH\left(-z - \sqrt{\mathrm{SNR}} \right)^{D-1}}{1 - H\left(-z - \sqrt{\mathrm{SNR}} \right)^{D}} \frac{e^{-\frac{(z+\sqrt{\mathrm{SNR}})^2}{2}}}{\sqrt{2\pi}} \tag{47}$$

In this regime we can make the approximation

$$H\left(-z-\sqrt{\mathrm{SNR}}\right)^{-D} \approx e^{DH\left(z+\sqrt{\mathrm{SNR}}\right)} \tag{48}$$

which allows us to approximate Eqn. 47 as

$$\begin{aligned} z &= -\frac{DH\left(-z-\sqrt{\mathrm{SNR}}\right)^{D-1}}{1-H\left(-z-\sqrt{\mathrm{SNR}}\right)^{D}}\frac{e^{-\frac{\left(z+\sqrt{\mathrm{SNR}}\right)^2}{2}}}{\sqrt{2\pi}} \\ &\approx -\frac{DH\left(z+\sqrt{\mathrm{SNR}}\right)}{H\left(-z-\sqrt{\mathrm{SNR}}\right)^{-D}-1}\left(z+\sqrt{\mathrm{SNR}}\right) \\ &\approx -\frac{DH\left(z+\sqrt{\mathrm{SNR}}\right)}{e^{DH\left(z+\sqrt{\mathrm{SNR}}\right)}-1}\left(z+\sqrt{\mathrm{SNR}}\right) \end{aligned} \tag{49}$$

where for $z+\sqrt{\mathrm{SNR}} \gg 1$ we used

$$H\left(z+\sqrt{\mathrm{SNR}}\right) \approx \frac{e^{-\frac{1}{2}\left(z+\sqrt{\mathrm{SNR}}\right)^2}}{\sqrt{2\pi}\left(z+\sqrt{\mathrm{SNR}}\right)} \tag{50}$$

From Eqn. 49 we see that for $DH\left(z+\sqrt{\mathrm{SNR}}\right) \ll 1$, $z \to -\frac{\sqrt{\mathrm{SNR}}}{2}$ and $f''(-\frac{\sqrt{\mathrm{SNR}}}{2}) \approx -1$. The saddlepoint approximation for the error is then

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz\left(1-H\left(-z-\sqrt{\mathrm{SNR}}\right)^{D}\right) \tag{51}$$

$$\approx \int_{-\infty}^{\infty} Dz e^{\log\left(1-e^{-DH\left(z+\sqrt{\mathrm{SNR}}\right)}\right)} \tag{52}$$

$$\approx e^{-\frac{\mathrm{SNR}}{8}}\left(1-e^{-\sqrt{\frac{2}{\pi\mathrm{SNR}}}De^{-\frac{\mathrm{SNR}}{8}}}\right)$$

This can be further approximated as

$$P_\epsilon \approx \sqrt{\frac{2}{\pi\mathrm{SNR}}}De^{-\frac{\mathrm{SNR}}{4}} \tag{53}$$

We can insert the SNR of overlaps into Eqn. 53 to determine the limits on the size of D for good decoding. For HRR the approximation for the error gives us

$$P_\epsilon \approx \sqrt{\frac{2L}{\pi N}}De^{-\frac{N}{4L}} \tag{54}$$

which implies that for an error of $\delta \ll 1$

$$\frac{N}{L} \approx 4\log\left(\frac{D}{\delta\sqrt{\pi}}\right) - 2\log 2\log\left(\frac{D}{\delta\sqrt{\pi}}\right) \tag{55}$$

From this, we expect the error to be small as long and SNR is large and D is polynomial, and not exponential, in N.

B. Decoding After Memory Retrieval

To analyze the effect of memory retrieval on the decoding error, we start by considering decoding from a degraded binarized structure which is a Hamming distance $\frac{m-1}{2}$ from the uncorrupted binarized structure. In this case, the effective SNR is found by making the replacement $\langle a_1 \cdot \hat{a}_1\rangle \to m$ in Eqn. 34. From this, we see that $\mathrm{SNR}(m)$ should

take the form

$$\mathrm{SNR}(m) = \frac{2}{\pi}\left(\frac{m}{\Sigma}\right)^2 \sim \frac{2m^2}{\pi}\frac{N}{L} \tag{56}$$

When $\hat{a}_1$ is decoded from an imperfectly retrieved structured from memory, we can instead make the replacement $\langle a_1 \cdot \hat{a}_1 \rangle \to \sqrt{c}m$. Here, $c \sim O(1)$ is a constant factor accounting for differences in overlap of the structure with relations within and outside the retrieval cue. Then $\mathrm{SNR}(m)$ takes the modified form

$$\mathrm{SNR}(m) \sim \frac{2cm^2}{\pi}\frac{N}{L} \tag{57}$$

The decoding error after retrieval from memory is then given by

$$\begin{aligned} P_\epsilon &\approx \int dm p(m) \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \sqrt{\mathrm{SNR}(m)}\right)^D\right) \\ &\approx \int dm p(m) \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \sqrt{\frac{2cm^2}{\pi}\frac{N}{L}}\right)^D\right) \end{aligned} \tag{58}$$

where $p(m)$ takes the same form as in Methods. For $N \to \infty$, $p(m) \to \delta(m - m^*)$ and the decoding error approaches

$$P_\epsilon \approx \int_{-\infty}^{\infty} Dz \left(1 - H\left(-z - \sqrt{\frac{2cm^{*2}}{\pi}\frac{N}{L}}\right)^D\right) \tag{59}$$

III. INSIGHTS FROM RANDOM PATTERNS

A. Network Order Parameters

For generic Hopfield networks, we can characterize the quality of memory retrieval by formally defining three network order parameters which quantify the overlap of the network state with the stored memories at each time step. The first is

$$m^\mu(t) = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^\mu \sigma_i(t) \tag{60}$$

which represents the overlap of the network state $\sigma(t)$ with the corresponding pattern σ^μ at time t. If the initial state of the network has an $O(1)$ with a small number of patterns, i.e., $m_0 = m^\mu(0)$, the memory retrieval process can be sufficiently described by including an additional order parameter

$$r^\mu(t) = \frac{1}{\alpha}\sum_{\nu \neq \mu}^{P} m^\nu(t)^2 \tag{61}$$

which represents the overlap of the state with all patterns except for μ. We can also define

$$m^{\mu 0}(t) = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^\mu(0)\sigma_i(t) \tag{62}$$

as the overlap of the state at time t with the initial state of the network after attempting to retrieve σ^μ where the overlap between $\sigma^\mu(0)$ and σ^μ is m_0. While m^0 is typically not considered for random patterns, it becomes relevant in the case of structured knowledge, where the retrieval cue can be constructed from a subset of relations rather than simply a corrupted version of a memory.

For large N, the distributions of order parameters over fixed retrieval conditions, i.e., $p(m)$, $p(r)$ and $p(m^0)$, are sharply peaked at the average values, denoted by $\langle m \rangle$, $\langle r \rangle$, and $\langle m^0 \rangle$. In the limit $N \to \infty$, these quantities depend solely on m_0 and α.

B. Overlap Distributions

As discussed in Methods, the empirical distribution $p(m)$ is bimodal and takes the general form

$$p(m) = (1 - p_1)p_{m<1}(m) + p_1\delta(m - 1) \tag{63}$$

where p_1 is the probability that a structure is perfectly retrieved from memory and $p_{m<1}(m)$ corresponds to the distribution of m for imperfectly retrieved memories. The shape of the distribution is characterized by p_1, the width of the lower m mode, σ_m and the mean of that mode, m^*. These quantities all depend on the initial overlap m_0 used to retrieve the memory Results for p_1, m^* and σ_m are shown in Fig. 2a, for several values of N and two values of α.

We also compare the overlap distributions for structures of length L with retrieval cue of length $L_0 = 2$ with the overlap distribution for a Pseudo-inverse network with random patterns for the corresponding values of m_0 given in Eqn. 35 in Fig. 3.

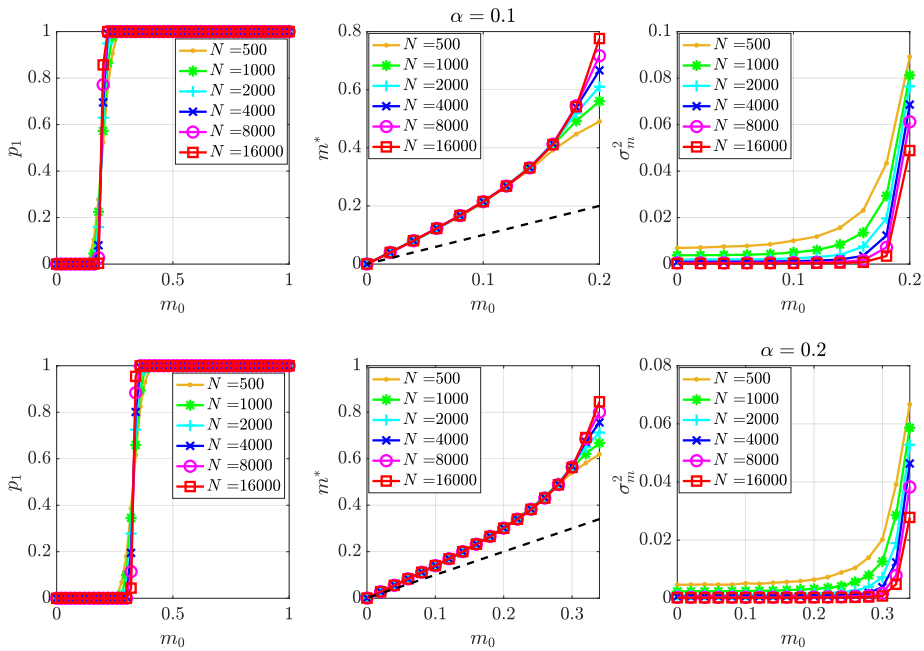

FIG. 2. p_1, m^*, and σ_m^2 are shown as functions of the initial overlap m_0 for several values of N for Pseudo-inverse networks storing random memories for two values of the memory load α. The dotted black line in the middle figures show that $m^* > m_0$ for these memory loads even when the initial state is far outside of the basin of attraction. $T = 20$ parallel updates are used for memory retrieval and averages are performed over 50 trials.

C. Retrieval Dynamics

When retrieving structures from memory (in the absence of noise), the update equation for the state of each neuron at time t is given by

$$\sigma_i(t) = \text{sgn}\left(\sum_j J_{ij}\sigma_j(t - 1)\right) \tag{64}$$

For all of the simulations in the main text, we consider parallel updates where all of the neurons $1 \leq i \leq N$. We find that serial updates give qualitatively similar results to parallel updates. In general, for large N we find

$$\langle m\rangle \sim f(\alpha)m_0 \tag{65}$$

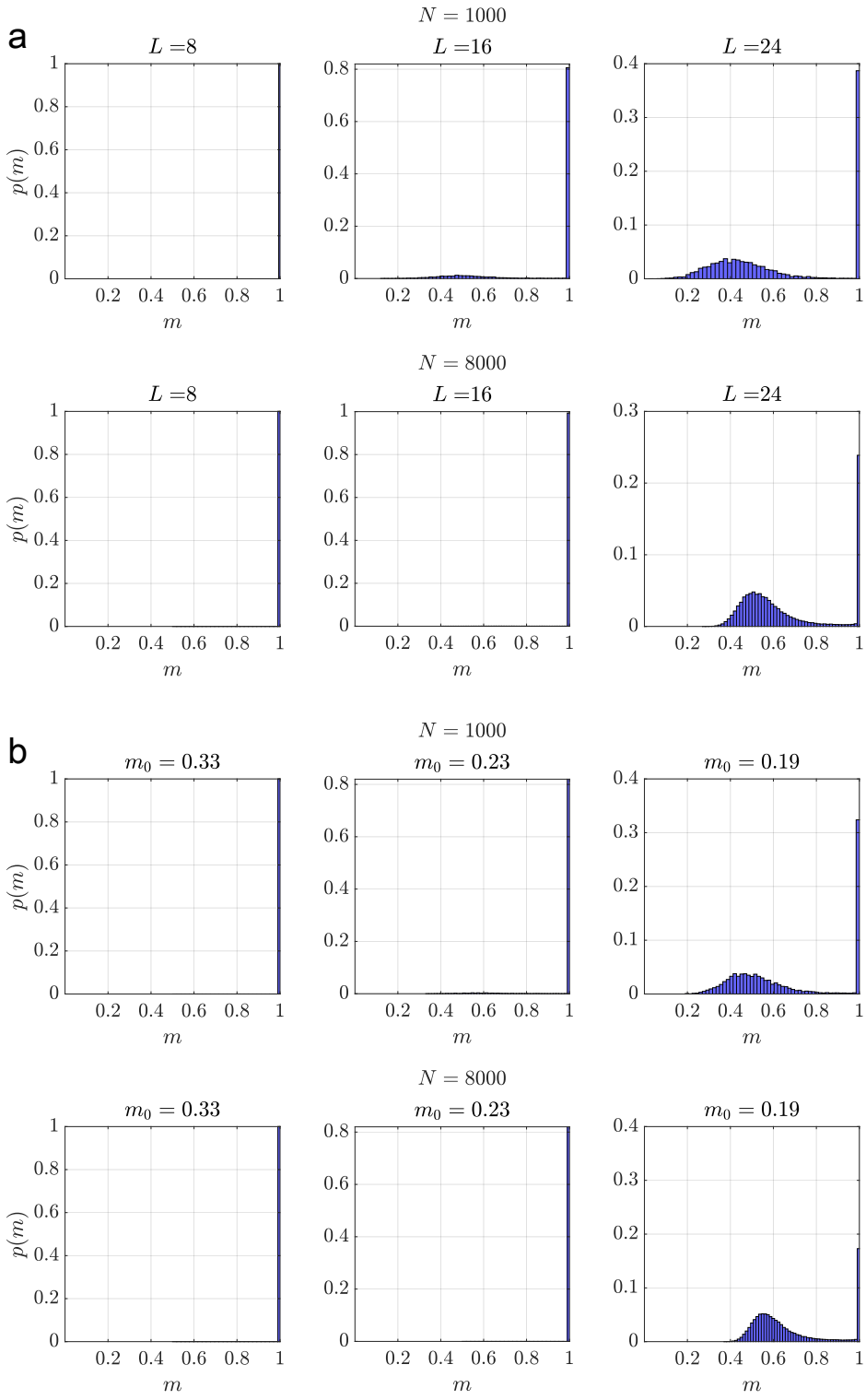

FIG. 3. (a) Histograms of the final overlaps m for structures of various lengths of L for $\alpha = 0.1$ and $L_0 = 2$ and 100 trials. $T = 20$ parallel updates are used to retrieve memories. The top panel shows $N = 1000$ and bottom shows $N = 8000$. (b) Histograms for values of m_0 obtained from Eqn. 37 corresponding to $L_0 = 2$ and $L = 8, 16, 24$ (left to right).

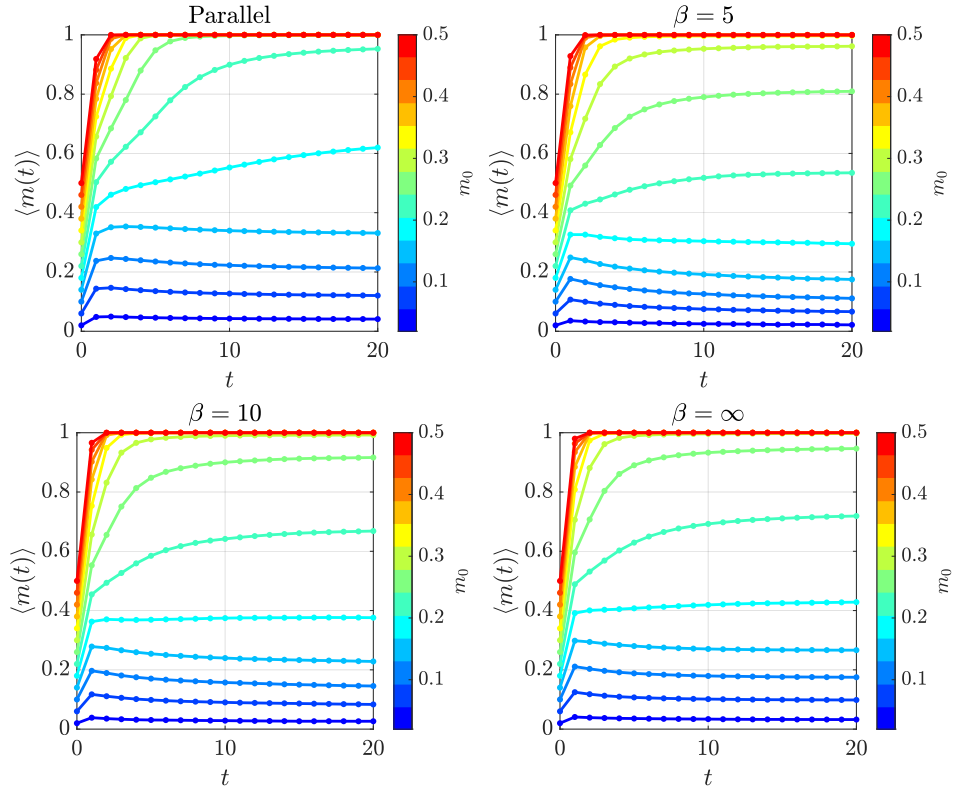

FIG. 4. Evolution of the averaged overlap order parameter $\langle m \rangle$ over $T = 20$ updates compared for parallel updates serial dynamics at zero noise ($\beta = \infty$), and serial dynamics with noise amplitude determined by inverse temperature $\beta = 5, 10$, for several different initial conditions m_0. The Pseudo-inverse rule is used to store the memories. $\alpha = 0.1$, $N = 1000$, and 50 trials are used for all figures.

where $\langle m \rangle \to m^*$ as $N \to \infty$. We find that serial updates obey the form in Eqn. 65 with a slightly smaller value of the coefficient $f(\alpha)$. Additionally, for serial dynamics, we can consider the robustness of our results under addition of noise in the updates. To do this, we use the Metropolis algorithm to obtain the final equilibrium state, where the amplitude of the noise is controlled by the inverse temperature β. At each update, the acceptance probability is given by

$$p\left(\sigma_i \to -\sigma_i\right) = \min\left(1, e^{-2\beta h_i \sigma_i}\right) \tag{66}$$

We see that Eqn. 66 becomes equivalent to Eqn. 64 for the noiseless case $\beta = \infty$. In Fig. 4, we show the average overlap $\langle m \rangle$ of the network state with the cued memory as a function of time for several different initial overlaps m_0 retrieved through parallel dynamics and serial dynamics for several different values of β. In Fig. 5 we show $\langle m \rangle$, $\langle r \rangle$, $\langle m^0 \rangle$ as function of retrieval cue overlap m_0. We see that the highest overlap is achieved for parallel updates when the serial updates are done in random order (alternative orders are discussed in [2]). However, the results are qualitatively similar for the different dynamics shown, demonstrating that small amounts of noise have little effect.

In Fig. 5a and c, we see that outside of the basin of attraction of the cued memory, i.e., values of m_0 where $\langle m \rangle < 1$, the final network state retains some memory of the initial state reflected by $\langle m^0 \rangle > m_0$.

D. Comparing Learning Rules

In the main text, we considered storing structured memories in recurrent neural networks where the synaptic weights are set via the Pseudo-inverse rule. This learning rule fully de-correlates linearly independent memories so that each memory is a perfect fixed point for $\alpha < 1$. However, the Pseudo-inverse rule is both non-local and non-incremental. This makes it unlikely to be biologically implemented in a straightforward manner.

We now consider two learning rules that are both local and incremental. The first is the Hebb rule and the second is the Storkey rule proposed in [3–5] (see Eqns. (26)–(28) in Methods). The Storkey rule can be viewed as a biologically

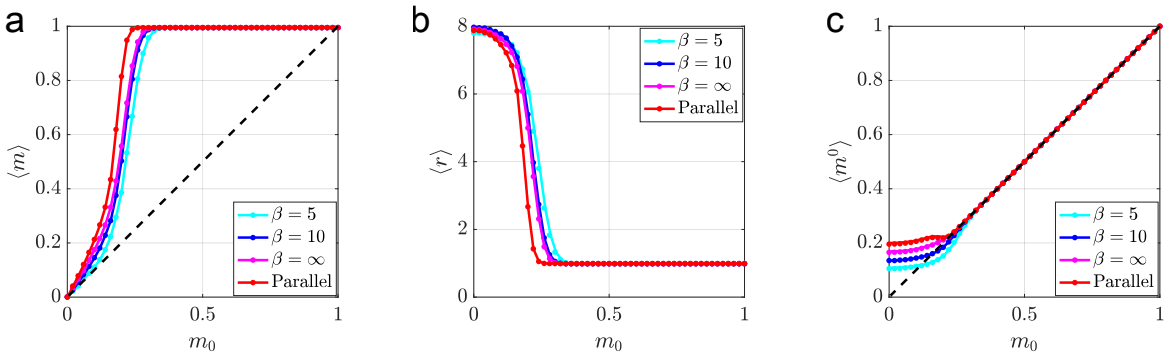

FIG. 5. Comparison of averaged order parameters $\langle m \rangle$ (a), $\langle r \rangle$ (b), and $\langle m^0 \rangle$ (C), after $T = 20$ updates as functions of retrieval cue overlap m_0 are compared for parallel updates and serial updates for various values of β. The dashed line in A and C is shown to compare the value of the final overlap with the cued memory $\langle m \rangle$ and the initial state $\langle m^0 \rangle$ to the initial overlap with the cued memory. The Pseudo-inverse rule is used to store memories and $\alpha = 0.1$, $N = 1000$, 50 trials are used for all figures.

plausible approximation to the Pseudo-inverse rule for small α [4]. As a result, the storage capacity of this rule $\alpha_c \approx 0.4$ is significantly higher than the Hebb rule ($\alpha_c \approx 0.14$) and the basins of attraction are larger and more even across different memories [5].

The storage and retrieval of random memories in Hopfield networks near saturation, i.e., when the number of memories P scaling linearly in N, is limited by interference between different memories [6]. The overlaps between different memories are characterized by the following matrix

$$C^{\mu\nu} = \frac{1}{N} \sum_{i=1}^{N} \sigma_i^\mu \sigma_i^\nu \tag{67}$$

Note that $C^{\mu\nu}$ is a $P \times P$ matrix that has the form of a sample covariance matrix.

We can compare the Pseudo-inverse, Hebb, and Storkey rules by looking at the forms of the local fields given by

$$h_i(t) = \sum_j J_{ij} \sigma_j(t) \tag{68}$$

From Eqn. 64, we see that a memory is a fixed point if $\sigma_i^\mu h_i > 0$ for all $i = 1, \ldots, N$. The elimination of the self-coupling term J_{ii} greatly increases the basins of attraction of the pattern. In the limit $N, P \to \infty$ this can be accomplished by replacing J_{ij} with $J_{ij} - \alpha\delta_{ij}$ [2]. The interference between memories is contained in the noise term in the local field.

We start by considering the Hebb rule, with synaptic weight matrix given by

$$J_{ij}^H = \sum_{\mu}^{P} \sigma_i^\mu \sigma_j^\mu, \ i \neq j \tag{69}$$

The local field for the Hebb rule can be expressed as

$$h_i^H(t) = \sum_{\mu=1}^{P} \sigma_i^\mu m^\mu(t) - \alpha\sigma_i(t) \tag{70}$$

Next we consider the the Pseudo-inverse rule with synaptic weight matrix expressed in terms of $C^{\mu\nu}$ as

$$J_{ij}^P = \sum_{\mu,\nu=1}^{P} \sigma_i^\mu (C^{-1})^{\mu\nu} \sigma_j^\nu, \ i \neq j \tag{71}$$

It is useful to decompose the state of the network $\sigma_i(t)$ in two parts as follows

$$\sigma_i(t) = \sum_{\mu=1}^{P} a^\mu(t)\sigma_i^\mu + \delta\sigma_i(t) \tag{72}$$

where $\delta\sigma_i(t)$ is orthogonal to all of the patterns ($\sum_i \sigma_i^\mu \delta\sigma_i(t) = 0$) and a^μ is related to the order parameter $m^\mu(t)$ via

$$a^\mu(t) = \sum_{\nu=1}^{P} (C^{-1})^{\mu\nu} m^\nu(t) \tag{73}$$

The local field for the Pseudo-inverse rule can then be expressed as

$$\begin{aligned} h_i^P(t) &= (1-\alpha)\sum_{\mu=1}^{P} a^\mu(t)\sigma_i^\mu - \alpha\delta\sigma_i(t) \\ &= (1-\alpha)\sum_{\mu,\nu=1}^{P} \sigma_i^\mu (C^{-1})^{\mu\nu} m^\nu(t) - \alpha\delta\sigma_i(t) \end{aligned} \tag{74}$$

We can see how Eqn. 74 suppresses the effects of overlaps by considering the state $\sigma_i = \sigma_i^1$. Then $a^\mu = \delta_{\mu 1}$ regardless of any correlations of the memories and $h_i^P = (1-\alpha)\sigma_i^\mu$. This implies that each memory is an eigenvector of J_{ij}^P with eigenvalue $(1-\alpha)$ so that all memories are perfect fixed points for $\alpha \geq 1$. By contrast, we see from Eqn. 70 that h_i^H becomes

$$h_i^H = (1-\alpha)\sigma^1 + \sum_{\mu=2}^{P} \sigma_i^1 m^\mu \tag{75}$$

contains additional noise terms of $O\left(\frac{1}{\sqrt{N}}\right)$ because of overlaps with the other patterns. This noise reduces the capacity of the network as well as the size of the basins of attraction for each memory which are both further reduced if the patterns are not random and contain correlations.
A simplified form of the Storkey rule, discussed in [4, 5, 7], is given by

$$\begin{aligned} J_{ij}^{S,0} &= 0 \\ J_{ij}^{S,\mu} &= J_{ij}^{\mu-1} + \frac{1}{N}\sigma_i^\mu\sigma_j^\mu - \frac{1}{N}\sigma_i^\mu h_j^\mu - \frac{1}{N}h_i^\mu\sigma_j^\mu \end{aligned} \tag{76}$$

where

$$h_i^\mu = \sum_{j=1}^{N} J_{ij}^{\mu-1}\sigma_j^\mu \tag{77}$$

Following the analysis in [7] We can find a more compact approximation for $J_{ij}^{S,P}$. We start with $P = 1$. Then $J_{ij}^{S,1}$ is given by

$$J_{ij}^{S,1} = \frac{1}{N}\sigma_i^\mu\sigma_j^\mu \tag{78}$$

For $P = 2$ we have

$$J_{ij}^{S,2} = \sum_{\mu=1}^{2} \frac{1}{N}\sigma_i^\mu\sigma_j^\mu - \frac{1}{N^2}\sigma_i^2 \sum_{k=1}^{N} \sigma_j^1\sigma_k^1\sigma_k^2 - \frac{1}{N^2}\sum_{k=1}^{N} \sigma_i^1\sigma_k^1\sigma_k^2\sigma_j^2 \tag{79}$$

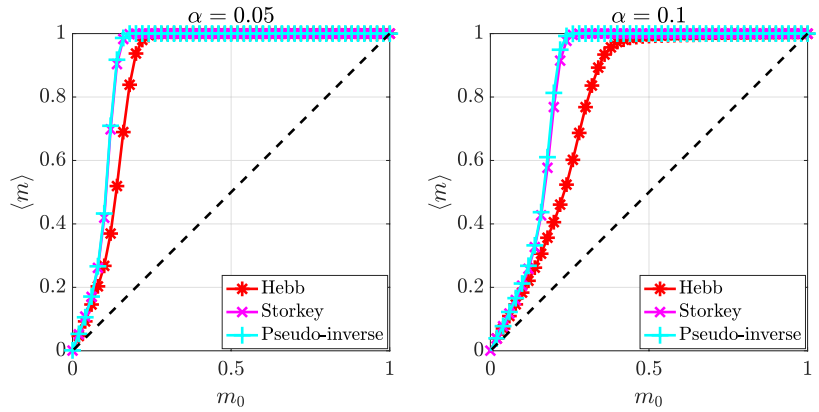

FIG. 6. Comparison of $\langle m \rangle$ as function of retrieval cue overlap m_0 for the Hebb, Storkey, and Pseudo-inverse learning rules for two values of α. $T = 20$ parallel updates, $N = 1000$ and 50 trials are used for both figures.

For P memories, keeping terms up to $O\left(\frac{1}{N^2}\right)$, J_{ij}^S can be expressed as

$$\begin{aligned} J_{ij}^S &\approx \frac{1}{N}\sum_{\mu=1}^{P}\sigma_i^\mu\sigma_j^\mu - \frac{1}{N^2}\sum_{\mu,\nu=1}^{P}\sum_{k=1}^{N}(1-\delta_{\mu\nu})\sigma_i^\mu\sigma_k^\mu\sigma_k^\nu\sigma_j^\nu \\ &\approx \frac{1}{N}\sum_{\mu=1}^{P}(1+\sum_{k=1}^{N}\left(\sigma_k^\mu\right)^2)\,\sigma_i^\mu\sigma_j^\mu - \frac{1}{N}\sum_{\mu,\nu=1}^{P}\sigma_i^\mu\left(\frac{1}{N}\sum_{k=1}^{N}\sigma_k^\mu\sigma_k^\nu\right)\sigma_j^\nu \\ &\approx \frac{2}{N}\sum_{\mu=1}^{P}\sigma_i^\mu\sigma_j^\mu - \frac{1}{N}\sum_{\mu,\nu=1}^{P}\sigma_i^\mu C^{\mu\nu}\sigma_j^\nu \end{aligned} \tag{80}$$

We can relate J_{ij}^S to the Pseudo-inverse rule given in Eqn. 71 by rewriting J_{ij}^P using the following series expansion of C^{-1}

$$\begin{aligned} C^{-1} &= (\mathbb{1} - (\mathbb{1} - C))^{-1} \\ &\approx 2\mathbb{1} - C + O(C^2) \end{aligned} \tag{81}$$

Plugging this expression back into Eqn. 71 gives us

$$J_{ij}^P \approx \frac{1}{N}\sum_{\mu=1}^{P}\sigma_i^\mu(2\delta^{\mu\nu} - C^{\mu\nu})\sigma_j^\nu \tag{82}$$

which is identical to the expression for J_{ij}^S given in the last line of Eqn. 80 up to terms of $O\left(\frac{1}{N^2}\right)$.

The expansion of C in Eqn. 81 converges if the eigenvalues of C are all contained in the interval $[0, 2]$. For random memories with components σ_i^μ drawn iid from $\mathcal{N}(0, 1)$, the distribution of the eigenvalues of $C^{\mu\nu}$ is given by the Marchenko-Pastur distribution given by, which talks the following form for $P, N \to \infty$

$$\rho(\lambda) = \frac{2}{\pi}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda} + \mathbb{1}\{\alpha < 1\}(1-\alpha)\delta(\lambda),\ \ \lambda_\pm = (\sqrt{\alpha} \pm 1)^2 \tag{83}$$

Since $\lambda \in [\lambda_-, \lambda_+]$, the expansion of C is valid for $\alpha < 0.17$ implying that $J_{ij}^S \approx J_{ij}^P$ for sufficiently small α and large N.

In Fig. 6 we compare $\langle m \rangle$ as a function of m_0 for two values of α for the three learning rules and in Fig. 7, we show m_{min} (as defined in the main text) as a function of α.

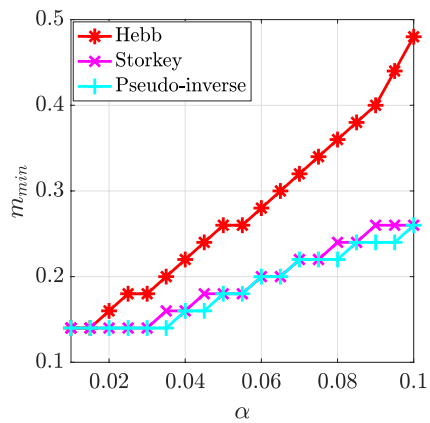

FIG. 7. The average size of the basins of attraction (as defined in the main text) for the Hebb Storkey and Pseudo-inverse learning rules are shown together for small values of α.

E. Connection to Storage of Knowledge Structures

While the various learning rules and dynamics have quantitative effects on the values on the various network order parameters at fixed memory load and network size, their qualitative behavior as a function of the retrieval cue overlap m_0 remains roughly the same for small values of α below capacity.

In general, we have found that outside of the basin of attraction, $\langle m \rangle$ (and m^*) scales linearly with m_0 for all three learning rules and takes the form in Eqn. 7 where the coefficient $f(\alpha)$ depends on the learning rule.

In all cases, for large N where $\langle m \rangle \to m^*$, the SNR behaves as

$$\begin{aligned} \text{SNR} &\sim \frac{2c}{\pi}\frac{(m^*)^2}{L} \\ &\sim \frac{8cf(\alpha)^2}{\pi^3}\frac{NL_0}{L^2} \end{aligned} \tag{84}$$

As a result, while the choice of learning rule and retrieval dynamics modify the value of $f(\alpha)$ defined in Eqn. 65 in the main text, they do not change the scaling behavior in Eqn. 85.

Comparing P_ϵ for structures stored using the Hebb, Storkey, and Pseudo-inverse rules in Fig. 8a. In general, we find that for low values of $\alpha \lesssim 0.3$, the error of decoding from memories stored via the Storkey rule is very similar to the Pseudo-inverse rule. Both are significantly lower the error obtained when decoding from memories stored using the Hebb rule. This suggests that the Storkey learning rule sufficiently decorrelates the different structures to allow for both efficient storage and retrieval of structured knowledge.

IV. TEMPORAL SEQUENCES AS SEQUENCES OF ATTRACTORS

Previously in [8, 9], it was shown that a temporal sequence of memories could be stored in a Hopfield network by adding a second asymmetric synaptic interaction of the form

$$J_{ij}^{(2)} = \frac{\lambda}{N}\sum_{\ell=1}^{L}\sigma_i^{\ell+1}\sigma_j^{\ell},\ i \neq j \tag{85}$$

to the synaptic weight matrix, where $L < P$ and $\lambda > 1$.

We can store Q sequences in the network by summing together a contribution $J_{ij}^{(2,q)}$ for each sequence, i.e.,

$$J_{ij}^{(2)} = \sum_{q=1}^{Q} J_{ij}^{(2,q)} \tag{86}$$

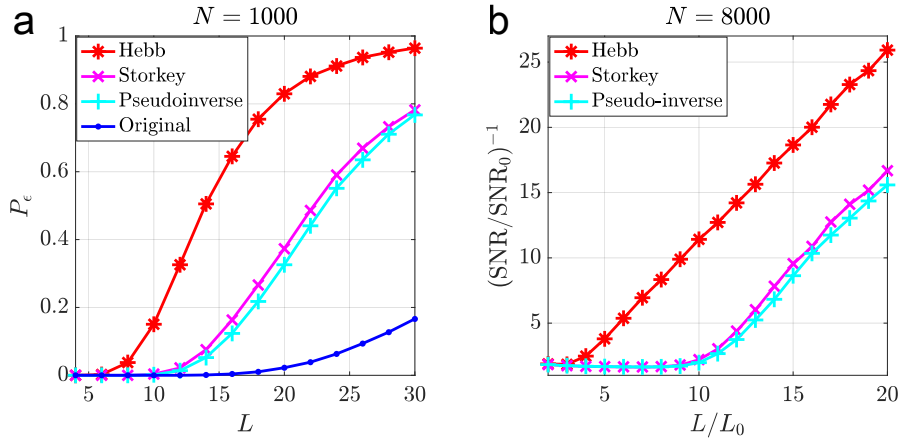

FIG. 8. (a) Comparison of the decoding error P_ϵ for structures of size $N = 1000$ stored using the Hebb rule, Storkey rule, and Pseudo-inverse rule at memory load $\alpha = 0.1$ and retrieved from memory with a partial cue of length $L_0 = 2$. The P_ϵ for the original structure is also shown for comparison and the size of the dictionary is fixed to $D = 30N$. (b) $(\mathrm{SNR}/\mathrm{SNR}_0)^{-1}$v. L/L_0 shown for $N = 8000$. $T = 20$ parallel updates are used for both figures and the is average is performed over $20,000$ memories.

as long as each memory is only contained in one sequence.

To retrieve a sequence, one starts with an initial state $\sigma_0 = \sigma(0)$ within the basin of attraction of the first memory in the sequence. The evolution of the network state is then given by

$$\sigma_i(t) = \mathrm{sgn}\left(\sum_j J_{ij}^{(1)}\sigma_j(t-1) + \sum_j J_{ij}^{(2)}\bar{\sigma}_j(t-1)\right) \tag{87}$$

where $\bar{\sigma}_i(t)$ is defined as

$$\bar{\sigma}_i(t) = \int_{-\infty}^{t} dt' w(t-t')\sigma_i(t) \tag{88}$$

where the function $w(t)$ represents a dynamic memory characterized by time constant τ. A simple choice for $w(t)$ uses the heaviside function, i.e. $w(t) = \frac{1}{\tau}\Theta(t-\tau)$, which we use for the simulation results shown in Fig. 6d of the main text. $J_{ij}^{(1)}$ is given by a learning rule for the conventional Hopfield network such as the Hebb, Storkey, or Pseudo-inverse rules.

For long sequences, transitions between attractors will be equally spaced with period t_0 after initial transients and the system will be in the pattern μ in the time interval $((\mu-1)t_0, \mu t_0)$. For a heaviside function t_0 is given by

$$t_0 = \frac{\tau}{2}\left(1 + \lambda^{-1}\right) \tag{89}$$

We can access the performance of this sequence memory by looking at $m^\mu(t)$ given in Eqn. 60 for each pattern in a sequence. Good performance requires $m^\mu(t) \approx 1$ during the interval $t \in ((\mu-1)t_0, \mu t_0)$ and $m^\mu(t) \ll 1$ for all other t.

Note that if a we store two sequences sharing items in the same memory network, transitions between attractors in the dynamics given in Eqn. 87 are no longer unique.

[1] T. A. Plate, Distributed representations and nested compositional structure, (1994).
[2] I. Kanter and H. Sompolinsky, Associative recall of memory without errors, Phys. Rev. A **35**, 380 (1987).
[3] A. Storkey, Increasing the capacity of a hopfield network without sacrificing functionality, in *Proceedings of the 7th International Conference on Artificial Neural Networks*, ICANN '97 (Springer-Verlag, Berlin, Heidelberg, 1997) p. 451–456.
[4] A. Storkey and R. Valabregue, A hopfield learning rule with high capacity storage of time-correlated patterns (1997).
[5] A. Storkey and R. Valabregue, The basins of attraction of a new hopfield learning rule, Neural Networks **12**, 869 (1999).
[6] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Storing infinite numbers of patterns in a spin-glass model of neural networks,

Phys. Rev. Lett. **55**, 1530 (1985).
[7] A. J. Storkey, Efficient Covariance Matrix Methods for Bayesian Gaussian Processes and Hopfield Neural Networks, , 138 (1999).
[8] H. Sompolinsky and I. Kanter, Temporal association in asymmetric neural networks, Phys. Rev. Lett. **57**, 2861 (1986).
[9] D. Kleinfeld and H. Sompolinsky, Associative neural network model for the generation of temporal patterns. Theory and application to central pattern generators, Biophysical Journal **54**, 1039 (1988).