CUE: A unified spiking neuron model of short-term and long-term memory

Jan Gosmann

Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada

Chris Eliasmith

Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada

Abstract

We present the context-unified encoding (CUE) model, a large-scale spiking neural network model of human memory. It combines and integrates activity-based short-term memory with weight-based long-term memory. The implementation with spiking neurons ensures biological plausibility and allows for predictions on the neural level. At the same time, the model produces behavioral outputs that have been matched to human data from serial and free recall experiments. In particular, well-known results such as primacy, recency, transposition error gradients, and forward recall bias have been reproduced with good quantitative matches. Additionally, the model accounts for the Hebb repetition effect. The CUE model combines and extends the ordinal serial encoding (OSE) model, a spiking neuron model of short-term memory, and the temporal context model (TCM), a mathematical memory model matching free recall data. To implement the modification of the required association matrices, a novel learning rule, the association matrix learning rule (AML), is derived that allows for one-shot learning without catastrophic forgetting. Its biological plausibility is discussed and it is shown that it accounts for changes in neural firing observed in human recordings from an association learning experiment.

Keywords: memory; modeling; computational neuroscience; spiking neural network; NEF

CUE: A unified spiking neuron model of short-term and long-term memory

Introduction

The study of short-term memory has given rise to the identification of a wide variety of behavioral effects, including primacy, recency, and forward bias. In addition, long-term memory effects have been observed in repeated short-term memory experiments, such as the Hebb repetition effect. Specifically, the primacy and recency effects in serial and free recall refers to the fact that test subjects are more likely to recall the first few (primacy) and last few items (recency) than the middle items in a memorized list. Forward bias highlights the tendency of subjects to recall items in forward order and nearby items together in free recall. Finally, the Hebb repetition effect is the observation that recall of a reoccurring list in a repeated serial recall experiment improves with the number of occurrences.

This variety of behavioral effect is only a small sample of the many memory effects described in the experimental literature. However, despite extensive behavioral characterization, it remains unclear how memory systems are neurally implemented such that they produce these behavioral effects. In fact, researchers have identified critical limitations in typical neural models that highlight the potential difficulty of making a unified short-term and long-term neural model. For instance, central among the open challenges is the stability-plasticity dilemma (Abraham & Robins, 2005). On the one hand, there is a need to quickly form new memories, sometimes even with a single exposure (also known as one-shot learning). Usually this is accomplished by having very high learning rates that result in rapid weight change. On the other hand, such high plasticity can easily lead to overwriting of old memories, making it difficult to capture long-term memory effects.

Regardless, given the importance and volume of research into memory, it is unsurprising that a large number of neural and non-neural memory models exists. However, viewed from the perspective of developing mechanistic, neuron-level models of

short-term and long-term memory, currently available models are generally limited. To begin, the majority of models are pure mathematical models that describe behavioral recall patterns, but leave it unclear how the proposed mechanisms can be implemented neurally (or even if a neural implementation is possible at all). As well, models that provide a neural implementation (e.g., Levy, Hocking, & Wu, 2005) often only focus on reproducing low-level neural findings, e.g., in the hippocampus, but do not connect to high-level behavior. Furthermore, many of these neural models use rate neurons as a convenient abstraction, but this neglects the need for robustness against noise introduced by discontinuous neural spikes. Moreover, rate neurons make it possible to use biologically questionable abstractions, such as backpropagation (e.g., Botvinick & Plaut, 2006). Finally, the majority of models do not address the important question of how changes in information flow can be accomplished during different phases of a memory task (e.g., presentation versus recall) without changing the model (i.e., only changing inputs).

We propose a new spiking neuron memory model that provides a unified account of a variety of short-term and long-term memory effects, including primacy, recency, forward bias, and Hebb repetition. We refer to this model as the context-unified encoding (CUE) model, and show that it addresses limitations of past modeling efforts by providing a low-level spiking neural network implementation that reproduces high-level behavioral data from memory experiments. The model combines activity-based short-term memory (STM) and weight-based long-term memory (LTM) to elucidate the interaction of long-term and short-term memory systems while matching data from a wider range of memory experiments.

We argue that this model provides new insight into the biological basis of cognition by: 1) providing an explicit mapping of cognitive function to spiking neural networks; 2) unifying long- and short-term memory systems into a single model; and 3) addressing how such a model can be controlled within the context of a behaving brain through simple manipulation of its inputs. We support these claims by demonstrating a variety of both

behavioral- and neural-level effects captured by the model that address observations related to both STM and LTM. Ideally such a model can be used to improve 'whole brain' models, such as Spaun (Eliasmith et al., 2012), and deepen our understanding of how memory systems support cognitive behaviour more generally construed.

To describe the CUE model, we begin by introducing the Neural Engineering Framework and Semantic Pointer Architecture methods that are used to construct it. We then provide an overview over the Ordinal Serial Encoding model and the Temporal Context Model as two prior models that this work is based on. Next a new learning rule, the association matrix learning rule (AML), is introduced, after which we provide an overview of the CUE model itself. We then present results that demonstrate that the AML can account for important neural observations, and that the full CUE model provides quantatitive accounts of a variety of behavioral data. We conclude with a discussion emphasizing the contributions and limitations of the proposed model.

Past models of memory

Perhaps the most influential conceptual description of working memory was proposed by Baddeley (1986). He suggested, based on experimental data, separate stores for visual and acoustic information, termed the *visuospatial sketchpad* and *phonological loop* respectively. These are controlled by a *central executive*. Furthermore, in Baddeley (2000) the model was extended with an episodic buffer for the binding of multimodal information and transfer to episodic long-term memory. While useful for characterizing the possible organization of (working) memory, and identifying the importance of the executive control, this conceptualization does little to elucidate mechanisms, and does not provide quantitative predictions about memory performance.

Some of these limitations are addressed by the plethora of mathematical models of memory that are available. Mathematical models tend to focus on making quantitative predictions at the behavioral level. Four such models include the perturbation model for

serial order by Estes (1972), the free recall model Search of Associative Memory (SAM) by Raaijmakers and Shiffrin (1981), the recognition memory model Retrieving Effectively from Memory (REM) by Shiffrin and Steyvers (1997), and the episodic memory model MINERVA2 by Hintzman (1988). The perturbation model is an early attempt to provide a mathematical framework for how remembered item positions can drift over time. In SAM, cues are assembled in a short-term memory to retrieve associations from a long-term associative memory. In the REM model, error prone copies of feature vectors derived from the study of items are stored. A recognition probe is matched to the stored feature vectors and a likelihood ratio of the match scores being generated by an old versus a new item is calculated. The MINERVA2 model, while focusing on episodic memory, also uses feature vectors that are stored as traces and can be probed by cues.

All of these models, and many others, either assume item-to-item or position-to-item associations. As well, starting with Anderson (1973), many models have used random, often high-dimensional feature vectors to represent individual items. Such an approach is broadly similar to the Semantic Pointer Architecture (SPA) that we adopt, and is discussed in more detail later. An especially influential model based on random feature vectors is TODAM2 (Murdock, 1993). It is able to fit a large body of experimental data and, in that regard, aims to be a general theory for item recognition, serial order, and associative memory. TODAM2 is of particular relevance to the model proposed here as it uses convolution for encoding associations. The SPA similarly uses circular convolution, and so is a direct descendant of TODAM2.

Another useful idea, that had its origin in mathematical models, is the idea of a randomly drifting context signal to which items are associated. Estes (1955) presented the first model of this type and Murdock (1997) extended the TODAM2 model in this way to explain additional data. The OSCAR model (Brown, Preece, & Hulme, 2000) uses a deterministic context signal that is generated from multidimensional oscillators to attempt to show how contexts can be reconstructed from a starting state. However, it does not

describe control mechanisms to determine how the context is re-instantiated to start recall, as we do below. As well, these past context-based models cannot explain the asymmetric conditional response probability (CRP) curves. CRP curves capture the probability of recalling items a certain distance (or lag) in a list from the current item. The temporal context model (Howard & Kahana, 2002), however, was specifically constructed to explain these asymmetric CRP free recall data. We incorporate core aspects of the TCM model here, and hence discuss it in more detail in a later section.

In addition to mathematical models, there have been several neural network, or connectionist, models of memory proposed, although they are less common. Many of these models focus on reproducing low-level findings in the hippocampus, such as sequence compression in replay (Levy et al., 2005) or place cells (Milford, Wyeth, & Prasser, 2004). The model presented by Hasselmo (2012), might be the most comprehensive hippocampus model to date, describing the storage of episodic memories as a spatial trajectory. A more recent model by Yu, Tang, Hu, and Tan (2017), constructs a three-layer spiking neural network that is able to encode a sequence over several iterations and replay it during a cycle of the theta rhythm. For these models, there is generally a large gap between what they explain and the high-level cognitive behaviour as modelled by mathematical models.

Nevertheless, there are some connectionist models that try to reduce this gap by addressing behavioral effects with a neural network implementation. For instance, Burgess and Hitch (1992) proposes a model for the articulary loop, Norman and O'Reilly (2003) for recognition and familiarity effects, and Botvinick and Plaut (2006) for immediate serial recall. All of these models use rate-based neurons as an abstraction. To the best of our knowledge there are only two memory-related models that use spiking neurons while at the same time connecting to behavioral data. The first is the ordinal serial encoding (OSE) model of serial recall by Choo (2010), which we extend here, and discuss further below. The second is a model of how serial lists might be stored within the hippocampus (Oliver Trujillo, 2014). It is able to reproduce neural data like replay and theta rhythm.

In the context of this past work, our goal with the CUE model is to address several current shortcomings. For instance, most models do not demonstrate how an executive system can interact with the model to switch phases (e.g. learning and recall), instantiate and update contexts, and generally coordinate memory function within a larger system. For instance, the majority of models we surveyed do not give an account of how responses are generated or how reinstantiation of recall context occurs. As well, most are not concerned with the interaction of short-term and long-term memory despite the importance for many fundamental effects on memory performance. As well, most models do not demonstrate that their proposed mechanisms are robust to implementation in noisy, spiking neural systems. Finite precision, limited numbers of neurons, and stochastic behavior at the neural level make some mathematical suggestions unlikely to be realized in the brain. For instance, TODAM2 requires unbounded dimensionality growth and precise calculation of convolutional powers (Choo, 2010).

We intend for the CUE model to begin addressing some of these concerns by combining activity-based short-term memory with weight-based long-term memory, while also specifying the required control and recall processes in spiking neural mechanisms. Biological plausibility is thus at least preliminarily addressed by implementing the model as a spiking neural network, using physiologically fixed parameters for single cells and synaptic interactions. However, despite this low-level implementation, it is validated against human, behavioral data, helping to close the gap between mathematical and neural descriptions.

Methods

To construct the spiking neural network CUE model, we use the Neural Engineering Framework. The processing and manipulation of the structures used to encode serial order is based on the Semantic Pointer Architecture. More specifically, we base the CUE model on the Ordinal Serial Encoding model (OSE; Choo, 2010) for activity-based short-term memory and the Temporal Context Model (TCM; Howard & Kahana, 2002) for

weight-based long-term memory. For synaptic weight changes, we derive the new association matrix learning rule. In this section we discuss each of these in turn.

The Neural Engineering Framework (NEF)

We use the Neural Engineering Framework (Eliasmith & Anderson, 2003) to construct spiking neural networks that implement specified dynamical systems. In this presentation of the NEF, we consider the representation and storage of a single item in memory. Notably this does not capture the functioning of the CUE model in any detail, but is rather intended to serve as a means of understanding the NEF methods in the current context. In the NEF, groups of neurons (*ensembles*) are used to represent real-valued, time-varying d-dimensional vectors $\boldsymbol{x}(t)$. For this example, let d be 32 dimensions and $\boldsymbol{x}(t)$ be the item to be stored. We assume that during presentation of the item, $\boldsymbol{x}(t)$ is a constant, after which $\boldsymbol{x}(t) = 0$ The spike train $a_i(t)$ of a neuron i in an ensemble representing $\boldsymbol{x}(t)$ is given by

$$a_i(t) = G\Big[\alpha_i \left(\boldsymbol{e}_i^\top \boldsymbol{x}(t)\right) + J_i^{\text{bias}}\Big] \tag{1}$$

where α_i is a gain factor, $\boldsymbol{e}_i$ is a random (unit-length) encoder or preferred stimulus (i.e. the firing rate will increase as $\boldsymbol{x}(t)$ aligns with $\boldsymbol{e}_i$), J_i^{bias} a bias current, and G the spiking neuron nonlinearity. The NEF supports many different spiking and rate neuron models, but here we use the common spiking leaky integrate-and-fire (LIF) model. Simply put, this equation describes how an input current (generated by stimulus $\boldsymbol{x}$) is turned into current which drives the spiking nonlinearity G. The LIF model itself provides a good balance of biological detail (i.e., spiking, having a membrane time constant) and minimal computational effort. Overall, the encoding in Equation 1 results in a distributed representation of the item across the neural ensemble. As is typical in neuroscientific studies, probing such a population with a variety of stimuli from the class of items will result in a 'tuning curve' for each neuron that characterizes which stimuli the neuron responds most strongly to. For example, such curves often generated by showing related

sets of items, such as faces, tools in later parts of visual cortex, or abstract shapes such as T intersections or swirls in earlier parts of visual cortex. Because it can be difficult to plot responses in a high- (e.g., 32-) dimensional space, such plots typically just show items and neuron response levels. Interestingly, we can also plot the response of the neuron along the encoder $\boldsymbol{e}_i$, reducing the response to a single dimension. This then captures the neuron response to increasing input in the preferred direction in a complex stimulus space.

The behavior of the neuron in this dimension is critical for characterizing its tuning. Specifically, the gain factor α_i and bias J_i^{bias} can be chosen to achieve a desired distribution of tuning curve intercepts and maximal firing rates within the range of represented values. That distribution determines how well the ensemble can compute various functions. In this model, the maximal firing rates are distributed uniformly between $200\,\text{s}^{-1}$ to $400\,\text{s}^{-1}$ (see Fig. 1). While this exceeds typical firing rates of actual in-vivo neurons, lower firing rates usually produce very similar results, but with more computational effort, if the neuron number is increased accordingly (e.g., Gosmann & Eliasmith, 2015).

Traditionally, a uniform intercept distribution is the default choice in the NEF. However, for d-dimensional representation with spiking neurons a higher accuracy can be achieved (Gosmann, 2018) by distributing intercepts according to the probability density function

$$p(x; d) = \frac{1}{B\left(\frac{1}{2}, \frac{d+1}{2}\right)} \cdot \left(1 - x^2\right)^{(d-1)/2}, \quad x \in [-1, 1] \tag{2}$$

where B is the beta function. Note that this distribution reduces to a uniform distribution for $d = 1$ (see Fig. 1). For the computation of some specific functions (e.g., rectification), other more specific intercept distributions are used.

From the spike trains a_i, the represented item vector $\boldsymbol{x}$ or some function $\boldsymbol{y} = f(\boldsymbol{x})$ of it can be linearly decoded after filtering with a synaptic filter $h(t)$. The synaptic filter models the post-synaptic current response in a dendrite upon the arrival of a spike. In essence, when a spike arrives at a synapse, it causes the release of neurotransmitters which then bind to the spine of the receiving neuron. This binding causes the opening of ion

channels with some time constant τ which allows ionic current to flow into the dendrite, essentially smearing the spike over a longer time period. The NEF uses this biophysical process to characterize temporal decoding of the spike trains generated by the above encoding. This temporal decoding is then weighted by connection weights to determine the input current to the receiving neuron. The NEF proposes how to factor these connection weights into a population decoder, and a population encoder (i.e. $\boldsymbol{e}_i$ above).

Considering only the decoders, we can write the function that is computed as:

$$\hat{\boldsymbol{y}}(t) = \sum_i \boldsymbol{d}_i^f \left[a_i * h\right](t) \tag{3}$$

where $\boldsymbol{d}_i^f$ are decoding weights for the function f. For the synaptic response $h(t)$, we use a simple decaying exponential $h(t) = \frac{1}{\tau}\exp(-t/\tau)$, which is a standard first-order model of a synaptic response. Critically, decoding is never assumed to occur in the biological system. Rather, decoding and encoding are theoretical constructs for characterizing neural information processing. Consequently, in the final model, only connection weights between neurons are presumed to exist. Mathematically, the connections weights are defined in terms of encoders and decoders (see Equation 5), but only the weights can be empirically measured.

The decoders themselves are determined by a regularized least-squares optimization minimizing the error

$$E^f = \int_{\boldsymbol{x}\in\mathcal{X}} \|f(\boldsymbol{x}) - \hat{\boldsymbol{y}}\| \, \mathrm{d}\boldsymbol{x}. \tag{4}$$

As a result, we are finding the linear optimal decoders for computing the desired function $\boldsymbol{y}(t)$. If we are simply attempting to store an item in memory, then this function should simply be the identity function (we just want to store what is shown), i.e. $\boldsymbol{y}(t) = \boldsymbol{x}(t)$. However the same optimization procedure can be used to approximate any function. Unlike standard learning methods (e.g. backpropagation), this optimization is guaranteed to have a global minimum and is efficient to compute. However, it does not optimize both the encoders and decoders simultaneously (which backpropagation does). In practice, a wide

variety of optimization methods can be used on NEF models (including backpropagation), depending on the constraints of the problem being modeled. Here, we only use the standard least squares method as it allows some model parameters (e.g. encoders) to be chosen in a manner informed by empirical data. We note, however, that the chosen optimization method is not thought to be biologically plausible. Instead, the resulting network is argued to be so because the final simulated system includes biophysically mapped mechanisms (including time constants, tuning curves, connection weights, etc.).

As mentioned previously, the neuron-to-neuron connection weights are a combination of decoders and encoders in the NEF. Specifically, the neural connection weights to transmit the represented vector from one ensemble to another ensemble are given by

$$\boldsymbol{W}_{ij} = \boldsymbol{e}_i^\top \boldsymbol{T} \boldsymbol{d}_j \tag{5}$$

where $\boldsymbol{T}$ gives a linear transform to apply to the transmitted vector. In the case of simply storing an item, T will be an identity.

Finally, the NEF allows us to implement dynamical systems. Given the state equations

$$\begin{aligned} \tfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t), \end{aligned} \tag{6}$$

with state matrix $\boldsymbol{A}$, input matrix $\boldsymbol{B}$, output matrix $\boldsymbol{C}$, feedthrough matrix $\boldsymbol{D}$ and characterizing the synaptic filter as above, the NEF shows that the neural implementation requires the input $\boldsymbol{u}(t)$ to the implementing ensemble to be multiplied with the synaptic time constant τ (of the implementing neurons), and a recurrent connection implementing the function $f(\boldsymbol{x}) = \tau\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}$. In the simple case of storing a representation over time, our dynamics are very simple (i.e., we do not want the representation to change when there is no input). That is, we can set $\boldsymbol{A} = 0$. As a result, the connection weights on the input to our ensemble will be equal to

$$\boldsymbol{W}_{ij} = \tau \boldsymbol{e}_i^\top \boldsymbol{d}_j, \tag{7}$$

and the connection weights on the recurrent connection of the ensemble will be

$$\boldsymbol{W}_{ii'} = \boldsymbol{e}_i^\top \boldsymbol{d}_{i'}. \tag{8}$$

Given these two sets of connection weights, the neurons in our ensemble will encode a 32-dimensional item vector while it is presented and then store it once the presentation ends (see Fig. 2). In essence, it will implement a very simple kind of working memory. More complex dynamical systems can also be implemented (Eliasmith, 2005), but are not required for the CUE model where more complex behaviour is achieved by gating the input to such simple working memory units.

The NEF allows us to specify that behavior at an abstract level (i.e. in terms of the dynamics of item representation $\boldsymbol{x}$(t)), and then generate the connection weights needed in a spiking network to implement that behavior. Critically, the final simulated models in the NEF consist only of spiking neurons, coupled through weighted post-synaptic currents whose time constants are determined by the neurotransmitter used in the synapse. In short, the simulated model implements a standard characterization of single neuron behavior. The function arises from the appropriate choice of connection weights. Nevertheless, the methods for finding encoders and decoders, the embedded dynamical systems, and the encoders and decoders themselves do not provide for directly measurable biological comparisons. Rather, the connectivity patterns, connection weights, time constants, neural firing patterns, behavior, and so on, provide directly empirically testable properties of the model. While the NEF is much more general than described here, this brief description is sufficient to capture the methods used to implement the main components of the CUE model.

The Semantic Pointer Architecture (SPA)

While the NEF enables us to represent and manipulate vectors with spiking neurons, the Semantic Pointer Architecture (Eliasmith, 2013) describes how vectors can be used to represent and manipulate structured, symbol-like information. It is an example of a vector

symbolic architecture (Gayler, 2004), in particular one implementing Holographic Reduced Representations (Plate, 1995) in spiking neurons.

In the SPA, random, high-dimensional vectors act as symbol-like representations and can be combined into more complex structures with certain mathematical operations. Element-wise addition is used as a superposition operation that produces a vector similar to both operands. Circular convolution, defined as

$$[\boldsymbol{x} \circledast \boldsymbol{y}]_i = \sum_{j=0}^{d-1} x_j y_{(i-j) \bmod d}, \tag{9}$$

is used to bind two vectors together, i.e., produce a vector that is dissimilar to both operands. Given a superposition of bound pairs $\boldsymbol{v} = \sum_i \boldsymbol{a}_i \circledast \boldsymbol{b}_i$, a pair's constituent $\boldsymbol{a}_k$ can be recovered given the other constituent $\boldsymbol{b}_k$ as

$$\hat{\boldsymbol{a}}_k = \boldsymbol{v} \circledast \boldsymbol{b}_i^+ = \boldsymbol{a}_k + noise \tag{10}$$

where $\boldsymbol{b}_k^+ = (b_{k,1}, b_{k,d}, b_{k,d-1}, \ldots, b_{k,2})^\top$ is the approximate circular convolution inverse. This inverse is a permutation (i.e., a linear transformation) of the original vector that, when multiplied by the original vector is approximately equal to the identity vector (i.e., a 1 followed by $d-1$ zeros). It is an approximation because the magnitude of the vector elements is not accounted for, except in the case of a unitary vector, when the approximate and exact inverses are the same. Effectively this inverse assumes that the vectors formed by permuting the original vector to compute circular convolution are orthonormal.

The SPA also contains additional elements (e.g., for action selection, vision, motor control, etc.), and techniques. In particular, representations, or "semantic pointers", in the system include many forms of compressed information not captured by the above operators. However, here we focus on the aspect of the SPA related to encoding structured conceptual information for serial working memory. Consequently, our presentation of SPA is tailored to that use.

The Ordinal Serial Encoding (OSE) model

The Ordinal Serial Encoding (OSE) model is a model of serial recall by Choo (2010). It is implemented as a spiking neural network using the NEF and SPA methods as well. The model output matches hallmark findings in serial recall such as the primacy and recency effects in the serial position curve as well as transposition gradients. The CUE model extends the OSE model with respect to activity-based short-term memory. Here we present an overview of the OSE to explain the CUE model extension.

In the OSE model, the n presented items are represented as Semantic Pointers $\boldsymbol{v}_i$ that are bound to position vectors $\boldsymbol{p}_i$. The position vectors are chosen randomly from the unit hypersphere, as is typical for non-perceptual representations in the SPA. Thus each position vector is unique for a specific list position. Because the overall list representation is the sum over bound item-position vectors, the accuracy of decoding positions in longer lists tends to degrade. A similar encoding of position has been shown to match human performance regarding list length and transposition effects (Choo, 2010). In the CUE neural implementation, these vectors are generated over time based on a signal that indicates a new item has been presented to the model (for details, see Gosmann (2018), section 7.2). Hence, they are intrinsic to the neural model, rather than being provided as an input, as in the original OSE model.

The bound item-position vectors are stored in two memory traces

$$\boldsymbol{m}_{\text{stim}} = \sum_{i=1}^{n} \gamma^{n-i} (\boldsymbol{v}_i \circledast \boldsymbol{p}_i) \tag{11}$$

$$\boldsymbol{m}_{\text{epis}} = \sum_{i=1}^{n} \rho^{n-i} (\boldsymbol{v}_i \circledast \boldsymbol{p}_i) \tag{12}$$

where $\boldsymbol{m}_{\text{stm}}$ and $\boldsymbol{m}_{\text{epis}}$ represent the short-term and an episodic, longer term memory store, respectively. Notably, in the final CUE model, the first of these is stored in the activities of an attractor network, constructed as described by Equation 6, while the second is replaced by the TCM-related component discussed below.

The adopted OSE encoding model contains two free parameters: a decay factor $\gamma < 1$

and a scaling factor $\rho > 1$. To recall an item $\hat{\boldsymbol{v}}_i$, the corresponding position vector is unbound as

$$\boldsymbol{v}_i \approx \hat{\boldsymbol{v}}_i = (\boldsymbol{m}_{\text{stim}} + \boldsymbol{m}_{\text{epis}}) \circledast \boldsymbol{p}_i^{+}. \tag{13}$$

The primacy and recency effect are obtained due to the differential effect of the decay and scaling factors γ and ρ.

In the original OSE implementation, both memory traces were stored in the activity-based representation in two spiking attractor networks (Eliasmith, 2005). This is plausible for the short-term storage of information and simplifies the model, but the episodic or long-term storage is generally assumed to depend on synaptic weight changes, typically considered to occur in the hippocampus (Eichenbaum, 2001). Thus, the CUE model replaces the episodic memory buffer by using a neural model of this subsystem that extends the TCM (described next) and uses synaptic weight changes. Furthermore, the CUE model implements the counting of positions $\boldsymbol{p}_i$ within spiking neurons, which was external to the original OSE implementation. This is realized with neural populations representing the different $\boldsymbol{p}_i$ vectors and a control mechanism that allows the model to advance the represented position by increasing the activity of the next neural population while inhibiting the population of the currently active position.

In sum, given an input list of items to remember, the CUE implementation of the OSE will internally generate a position vector, which it binds to the item vector, and then stores the resulting bound vector in a recurrent neural network memory. In Figure 6, this is the function of the OSE component.

The Temporal Context Model (TCM)

The temporal context model (TCM; Howard & Kahana, 2002) is a mathematical memory model, being applied to matching data from immediate, delayed, and continual distractor free recall tasks. In particular, it reproduces the tendency of humans to recall items in a forward order and to recall items together that were nearby in the learned list.

The model was proposed as a single store model, but Davelaar, Usher, Haarmann, and Goshen-Gottstein (2008) argued that the TCM best explains episodic or longer-term memory, and a separate short-term store is still needed. We adopt the same view here, which is also supported by fMRI data (Talmi, Grady, Goshen-Gottstein, & Moscovitch, 2005), where hippocampal areas were only activated for the retrieval of items presented early in a list. As the delayed and continual distractor recall conditions use a distractor task designed to prevent active rehearsal, it is plausible to assume that the TCM addresses more long-term synaptic storage, compared to the OSE model.

The TCM assumes, like many other memory models, a time-varying context signal that items are associated with. In contrast to those other models (e.g. Estes, 1955; Murdock, 1997), the context signal does not evolve randomly, but depends on the presented and recalled items. Both the items $\boldsymbol{f}_i$ and the context signal $\boldsymbol{c}$ are represented as vectors. The TCM constrains the items $\boldsymbol{f}_i$ to be orthogonal, but for the CUE model, we relax the orthogonality constraint to almost (instead of perfectly) orthogonal vectors (as others have done, c.f. Sederberg, Gershman, Polyn, and Norman (2011)), to enable the natural usage of Semantic Pointers. Furthermore, $\boldsymbol{f}_i$ includes both item $\boldsymbol{v}_i$ and position $\boldsymbol{p}_i$ information in the CUE model.

In the TCM, associations between items and contexts are stored in two outer-product association matrices. The outer product matrix

$$\boldsymbol{M}^{\mathrm{CF}} = \sum_i \boldsymbol{f}_i \boldsymbol{c}_i^{\top} \tag{14}$$

gathers the associations from context vectors to items. This matrix can be easily updated by adding further outer products of context/item pairs. The reverse associations from items to context vectors are stored in $\boldsymbol{M}^{\mathrm{FC}}$. When an item is presented or recalled, it is used to retrieve the associated context $\boldsymbol{c}_i^{\mathrm{IN}} = \boldsymbol{M}^{\mathrm{FC}} \boldsymbol{f}_i$, to update the current context $\boldsymbol{c}_i$ according to the *evolution equation* (Fig. 3)

$$\boldsymbol{c}_i = \theta_i \boldsymbol{c}_{i-1} + \beta \boldsymbol{c}_i^{\mathrm{IN}}, \tag{15}$$

where β is a free parameter controlling the rate of the context drift and $0 < \theta_i \leq 1$ is given by

$$\theta_i = \sqrt{1 + \beta^2 \left[\left(\boldsymbol{c}_{i-1}^{\top} \boldsymbol{c}_i^{\mathrm{IN}} \right)^2 - 1 \right]} - \beta \left(\boldsymbol{c}_{i-1}^{\top} \boldsymbol{c}_i^{\mathrm{IN}} \right) \tag{16}$$

to keep $\boldsymbol{c}_i$ at unit length in each time step. However, this is not strictly necessary for a good data match in the CUE model, so to simplify the neural implementation, we fix θ_i to the asymptotic value for $\left(\boldsymbol{c}_{i-1} \boldsymbol{c}_i^{\mathrm{IN}} \right) \to 0$ given by

$$\theta_i = \sqrt{1 - \beta^2}. \tag{17}$$

In the TCM, a θ_i close to this value is used whenever the item i has not been presented for a sufficiently long time, so that the context $\boldsymbol{c}_i^{\mathrm{IN}}$ has become orthogonal to the current context.

The asymmetric bias to forward recall observed in humans is introduced into the model by Equation 15. The similarity of contexts $(\boldsymbol{c}_i \boldsymbol{c}_j)$ is symmetric for the lag $j - i$, but $\boldsymbol{c}_i^{\mathrm{IN}}$ gets only included in context vectors $\boldsymbol{c}_j$ with $j \geq i$ (Fig. 4).

While $\boldsymbol{M}^{\mathrm{CF}}$ is updated by simply adding in new outer products, the TCM specifies a separate updating equation for $\boldsymbol{M}^{\mathrm{FC}}$, to achieve a specific contribution of $\boldsymbol{c}_i^{\mathrm{IN}}$ and $\boldsymbol{c}_i$ despite those two contributions not necessarily being orthogonal. Again, a good match to the data and simpler neural implementation can be achieved by relaxing this constraint and directly adding the outer products with a fixed scaling parameter b according to

$$\boldsymbol{M}_{i+1}^{\mathrm{FC}} = \boldsymbol{M}_i^{\mathrm{FC}} + b \boldsymbol{c}_i \boldsymbol{f}_i^{\top}. \tag{18}$$

To recall an item, a mixture of associated items is retrieved from the current context as $\hat{\boldsymbol{f}}^{\mathrm{IN}} = \boldsymbol{M}^{CF} \boldsymbol{c}$ and some form of cleanup to a single item is performed. The recalled item can then be used to recall an associated context $\boldsymbol{M}^{\mathrm{FC}} \boldsymbol{f}^{\mathrm{IN}}$, update the current context, and recall further items. Different cleanup strategies can be used. In the original TCM, the probability of retrieving an item $\boldsymbol{f}_i$ was obtained with Luce's choice rule as

$$P(\boldsymbol{f}_i | \boldsymbol{f}^{\mathrm{IN}}) = \frac{\exp\left(\frac{2a_i}{\tau}\right)}{\sum_j \exp\left(\frac{2a_j}{\tau}\right)} \tag{19}$$

with a set of activities $a_i = \boldsymbol{f}_i^\top \boldsymbol{f}^{\text{IN}}$ and a sensitivity parameter τ. A more recent version of the TCM presented by Sederberg, Howard, and Kahana (2008) uses a more biological plausible winner-take-all process known as the leaky, competing accumulator model (Usher & McClelland, 2001). However, Gosmann, Voelker, and Eliasmith (2017) showed that this cleanup process can be problematic to integrate in larger scale neural models because it tends to be slow to converge to a specific winner as more options become available, and the distinctions between them thus become noisier. Thus we use the independent accumulator in the CUE model.

In Figure 6 the three components on the left-hand side make up the TCM-related elements of the CUE model. These components use the input items to recall and update a context representation that is tracked during encoding and recall. This function is preserved in CUE, although the inputs and mechanisms are different than for the original TCM. For instance, item-to-context and context-to-item associations are used, as in TCM, for encoding information in the memory, but the updates necessary for these associations are learned using a spike-based learning rule, described next.

The Association Matrix Learning Rule

Due to the spiking neural implementation of the CUE model, outer products cannot simply be added to the association matrices $\boldsymbol{M}^{\text{FC}}$ and $\boldsymbol{M}^{\text{CF}}$. Instead a learning rule, the Association Matrix Learning rule (AML), is required to determine the appropriate synaptic weight changes. We propose and derive the rule here, to allow for online learning of the matrices as needs to occur in a biological system.

In the NEF, the connection weights $\boldsymbol{W}$ between two neural ensembles that implement a linear transform, such as an association matrix $\boldsymbol{M}$, after learning n associations can be expressed as:

$$\boldsymbol{W}_n = \boldsymbol{E}\boldsymbol{M}_n\boldsymbol{D} = \boldsymbol{E}\left(\sum_{i=1}^{n} \boldsymbol{v}_i \boldsymbol{u}_i^\top \boldsymbol{D}\right) \tag{20}$$

where $\boldsymbol{E}$ is the matrix of encoders $\boldsymbol{e}_j$ (one for each of the j post-synaptic neurons) of the

post-synaptic ensemble and $\boldsymbol{D}$ is the matrix of decoders $\boldsymbol{d}_k^f$ (one for each of the k pre-synaptic neurons) for the identity function $f(\boldsymbol{x}) = x$ of the pre-synaptic ensemble. From the equation we can see that as we progress through the n items, all weight changes can be described as decoder changes given by

$$\begin{aligned} \tilde{\boldsymbol{D}}_{i+1} &= \tilde{\boldsymbol{D}}_i + \Delta\tilde{\boldsymbol{D}}_i \\ \Delta\tilde{\boldsymbol{D}}_i &= \boldsymbol{v}_i \boldsymbol{u}_i^\top \boldsymbol{D} \end{aligned} \tag{21}$$

where $\tilde{\boldsymbol{D}}$ is the matrix of learned decoders. Because neural activity evolves continuously in time, this equation of discrete updates has to be converted to continuous time form:

$$\frac{\mathrm{d}\tilde{\boldsymbol{D}}}{\mathrm{d}t} = \eta \boldsymbol{v}(t)\boldsymbol{u}(t)^\top \boldsymbol{D} \tag{22}$$

with learning rate η, to be usable within an online, continuously learning network. Note that in this formulation not all associations are necessarily added with the same strength, but the learning rate and presentation time determine the association strength.

The AML is closely related to the previously proposed prescribed error sensitivity (PES) learning rule (Bekolay, Kolbeck, & Eliasmith, 2013; MacNeil & Eliasmith, 2011). In fact, the only difference is the inclusion of the $\boldsymbol{D}$ matrix. This makes it possible, in contrast to PES, to learn associations in a one-shot fashion (Fig. 5). Specifically, consider the AML written in terms of presynaptic firing:

$$\frac{\mathrm{d}\tilde{\boldsymbol{D}}}{\mathrm{d}t} = \eta \boldsymbol{v}(t)\boldsymbol{a}_u(t)^\top \boldsymbol{D}^\top \boldsymbol{D} \tag{23}$$

This is identical to the PES rule (and similar to other error driven Hebbian rules), except for the inclusion of the $\boldsymbol{D}^\top \boldsymbol{D}$ term. This term computes the correlation of the decoding between the presynaptic neurons, effectively weighting the update by that correlation. This means that when neurons are correlated in their response to a cue, they will increase their responsiveness together. This results in a much faster convergence of a select group of cells onto the cue, allowing them to generate the desired output, without adversely affecting other groups of cells correlated to a different cue. Consequently, there is less interference

between groups of cells than there would be without the correlation matrix. Indeed an identity correlation matrix (as is assumed by the PES rule), would allow all cells to learn to respond to all inputs, effectively overwriting old cues with new ones, leading to catastrophic forgetting.

The CUE model aims to be biological plausible and therefore we consider the biological plausibility of the AML. In the formulation above, the learning rule requires some form of 'weight sharing,' as $\boldsymbol{D}$ is the same decoder matrix that is used to decode $\boldsymbol{u}(t)$. Weight sharing is well-known to be an "unbiological" assumption. However, assuming we obtain $\boldsymbol{u}(t)$ from the pre-synaptic learning ensemble with neuron activities $\boldsymbol{a_u}(t)$, the learning rule can be written as

$$\frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = \eta \boldsymbol{E}\boldsymbol{v}(t) \left(\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{a}_{\boldsymbol{u}(t)}\right)^\top. \tag{24}$$

This no longer requires weight sharing, but still requires a symmetric weight matrix from the pre-synaptic ensemble for the calculation of the weight change. Ensuring symmetry can be accomplished with a biologically plausible local learning rule (Gosmann, 2018). A related result by Hua, Houk, and Mussa-Ivaldi (1999) demonstrates the emergence of symmetric weights with Hebbian learning. More interestingly, the problems of weight-sharing or symmetric decoder matrices can be completely avoided if $\boldsymbol{D}^\top \boldsymbol{D}$ happens to be the identity matrix (or at least reasonably close). That could be achieved if the neurons of the pre-synaptic ensemble use a sparse representation. Such sparsity allows for simple Hebbian learning from one set of sparsely firing neurons to the neurons representing the target because, for other cues, different sets of neurons will fire. It is worth highlighting that such sparse firing is observed in the dentate gyrus of the hippocampus (Rolls, 2013). However, we have not enforced sparse firing in the current model. Consequently improving the locality of the AML remains for future work, using either the enforcement of sparsity, or the local symmetric rule, or possibly both.

This learning rule is not a separate component of the CUE model, but rather provides a means of using spike-based mechanisms to update the association matrices core

to TCM. That is, the context-to-item and item-to-context associations are captured by learning these relations over time in the connection weights modified by AML.

The context-unified encoding (CUE) model

In the context-unified encoding model we extend the OSE and TCM to address the interaction of activity-based and synaptic memory as well as bridging from the level of spiking neurons to behavioral data of serial and free recall in a single unified model. We use the $\boldsymbol{m}_{\text{stm}}$ memory trace of the OSE model, but replace $\boldsymbol{m}_{\text{epis}}$ with an extended TCM implementation in spiking neurons. As $\boldsymbol{m}_{\text{stm}}$ requires position cues, the long-term memory component of the TCM should be able to provide such cues. Hence, we store the combination of the list item $\boldsymbol{v}_i$ and position vector $\boldsymbol{p}_i$ as $\boldsymbol{f}_i = \boldsymbol{v}_i + \boldsymbol{p}_i$ in the TCM associations. Notably, if the position information is not included, the serial working memory will not work properly, as it relies on position information for correct recall, especially if the same item appears multiple times in the list. If the item information is not included, free recall will be difficult as position would then fully define the context. If our input was a convolution of position and item vectors (as opposed to superposition that we use here), then position or item information (alone) would not be sufficient to generate the appropriate context, which would make recall no longer function. Superposition is important in the case where either element might be used later to recall the trace (convolution is important where the exact association between elements needs to be encoded).

Figure 6 gives a high-level overview of the structure and information flow in the CUE model. Note that each functional part consists of multiple groups of spiking neurons connected to perform its specific tasks. Furthermore, the figure does not show any routing and control related structures in the model for clarity, although the model also implements these in spiking neurons. Due to the complexity of the model, we do not discuss all implementational details here and restrict ourself to higher-level explanations. However,

the model source code providing all the specifics is available online for the interested reader[1], as is a comprehensive description of the model (Gosmann, 2018).

The short-term memory component consists of the *OSE* and *position* networks, which we take to be housed in memory-related cortical structures, like the dorsolateral prefrontal cortex, which is thought to be related to working memory (Hoshi, 2006; Ma et al., 2011). The *position* network keeps a representation of the current list position $\boldsymbol{p}_i$ active, and increments the position given an appropriate control signal. The accuracy of this representation is independent of the position, but the total number of representable positions is limited. The position $\boldsymbol{p}_i$, and the current list item $\boldsymbol{v}_i$, are inputs to the *OSE* network. The *OSE* network binds these two semantic pointers with circular convolution and adds them to the $\boldsymbol{m}_{\text{stm}}$ memory trace stored in a high-dimensional attractor network. The *position* input is also used to retrieve items from the short-term memory by unbinding the position from the memory trace and feeding the result to the *item recall* network.

The episodic memory component is constituted mainly of the *context*, $\boldsymbol{M}^{\text{FC}}$, and $\boldsymbol{M}^{\text{CF}}$ networks. In this component, the current context is stored by the *context* network and is updated for each new input. The input is provided from the $\boldsymbol{M}^{\text{FC}}$ network that recalls the associated context for an item. The item input to $\boldsymbol{M}^{\text{FC}}$ is the current list item and position vector during the presentation phase, and the last recalled item and position during the recall phase. Furthermore, there is a connection back from *context* to $\boldsymbol{M}^{\text{FC}}$ to allow an association from $\boldsymbol{f}_i$ to the current context to be created. The $\boldsymbol{M}^{\text{CF}}$ network receives the context as input to recall the associated items and positions. During serial recall, however, it directly receives the context recalled by the $\boldsymbol{M}^{\text{FC}}$ network. To learn these associations with the AML, the presented list items and current position vector are an additional modulatory input.

Together the short term and episodic memory elements of the model are what store information over time during a memory task. They are storing similar information (i.e.,

[1] https://github.com/ctn-archive/cue-model

position and item), but in quite different ways. The episodic memory system is embedding sums of position and item memories in connection weights that associate them with the current context. The short term memory system is binding position and item representations and storing them in neural activities in a recurrent network. This combination of using sums and bound representations, allows item-position representations to be flexibly used for serial recall and free recall, depending on the demands of the task.

In either case, during recall, the recalled noisy Semantic Pointers need to be "cleaned up", that is, associated to the closest previously known vector. This is done by independent accumulator (IA) networks with a thresholding layer (Gosmann & Eliasmith, 2017). The recall networks for items and positions are separate, as shown in Figure 6, but they function in the same manner. In both cases, a constant bias is integrated to a threshold in an accumulator dimension to put a time limit on recall attempts. The same signal causes the network to start with a new attempt if the current item cannot be recalled. Input to both recall networks is provided from the TCM component, more precisely $\boldsymbol{M}^{\mathrm{CF}}$, but the *OSE* component only projects to the item recall network. This is because position information is used to define the context, which is not explicitly stored in the *OSE* component. Gaussian noise is added to the inputs to account for external influences not explicitly modeled. Once an item or position has been recalled, it gets added to an attractor network storing recently recalled items, and inhibits these to prevent repetition errors (which are rare in humans). Specifically, the output of this attractor network is subtracted from the memory trace before being considered by the IA network. As a result, the independent accumulator network considers only those responses that have not yet been produced. The output of the main recall network (after the independent accumulator) is regarded as the model's output. The position recall output is used to update the *position* network in free recall. This allows recalled information from the episodic memory to facilitate short-term recall.

In summary, there are three core components of the CUE model that work in concert to produce its behaviour. These are the short term (OSE) component, the long term

(TCM) component, and the recall (IA) component. As mentioned, the short and long term components construct differing representations that provide for flexible memory behavior. The recall component is used to decide what items to output as a function of these representations, task demands, and recently reported items. During a given memory task, these components must be coordinated in different ways using plausible neural mechanisms.

Control in the CUE model. To achieve the desired model behavior, the flow of information has to be controlled appropriately in the CUE model. This is done on multiple levels by gating signals with inhibition based on the input signals. Note that this is done without changing any connectivity in the model. Crucially, this means that the CUE model can be used in an online manner, and as part of a larger cognitive model in a straight forward manner. Only simple control (e.g., whether to encode or recall) and content inputs (e.g., the current list item) need to be provided, there is no need to restructure the model, intervene during task performance, or move signals by hand when switching from free to serial recall, or from encoding to recall phases of a task.

The general task (like free or serial recall) and task phase (presentation, distractor, and recall phase) are used to control the 'effective' connectivity. For instance, during encoding, specific routing is employed for each item until it has been fully stored in memory. Furthermore, some local control of information routing happens in the individual networks to have them perform their desired functions. One example are the $\boldsymbol{M}^{\mathrm{FC}}$ and $\boldsymbol{M}^{\mathrm{CF}}$ networks inhibiting the modulatory input signals when the desired association strength is reached.

The overall information routing during different task phases is shown in Figure 7. During the presentation phase, parts of the recall networks and their inputs are inhibited. These networks are disinhibited in the recall phase, while the item input is absent. Learning of new associations is disabled during the recall phase by inhibiting the learning rule's $\frac{\mathrm{d}\tilde{\boldsymbol{D}}}{\mathrm{d}t}$ signal (given in Eq. 22). During serial recall, the current position $\boldsymbol{p}$ is used to decode an item from the OSE store and to retrieve the associated context with the $\boldsymbol{M}^{\mathrm{FC}}$

weights. The retrieved context is used to directly retrieve associated items via $\boldsymbol{M}^{\mathrm{CF}}$ and feed them to the recall network. During free recall, the routing differs slightly. Items are retrieved in the current context, not the retrieved context. However, retrieved context vectors update the current context according to the evolution equation. Furthermore, a successfully recalled position updates the position $\boldsymbol{p}$ to allow recall contributions from the OSE without iterating over positions in a fixed order.

Within each phase, a more fine-grained control of the information flow in some of the model networks is required. The corresponding control signals are generated with the network shown in Figure 8. Each presented or recalled item is fed into an attractor network with a slow synaptic time constant. A dot product is computed between this network and the item that increases depending on how long the item has been present. This signal is thresholded, and serves as basis for multiple control signals that differ slightly between the presentation and recall phases.

We describe the recall phase first. First, the rising flank of the signal is used to trigger the increment of the current position $\boldsymbol{p}$. Second, while the signal is below threshold, the current context representation gets transferred to a secondary buffer. Once the signal exceeds the threshold, this secondary buffer, and newly retrieved context for the item, are combined and used to overwrite the current context. Given these signals, item and position information is always recalled in parallel.

During the presentation phase, these two control signals are inverted. The effect is that during the presentation phase the context is updated as soon as a new item is presented, while the position is not incremented until it has been stored with the current position associated. During the recall phase, the context is updated with a slight delay after a successful recall to allow the network to fully process the recall and use the context retrieved from the updated position for the update. The position increment happens immediately during the serial recall phase. The immediate threshold of the dot product is further used independent of the current phase to control when new items are added into

the OSE memory representation and to signal the recall networks when sufficient time has passed for processing the currently recalled item to start the next recall.

Additional control considerations rise in the case of distractor tasks. As these are irrelevant to the main memory tasks, the learning of associations with synaptic-weight changes is deactivated during the distractors. The OSE short-term memory is still influenced by these items, but uses a special Semantic Pointer for the position indicating an irrelevant item. In other words, we hypothesize that distractors are essentially "marked" as distractors during the task, but still influence the memory trace.

Overall, using these kinds of inhibitory control structures is consistent with current understanding of cortical control through the cortex-basal ganglia-thalamus-cortex loop (Eliasmith, 2013). In particular, thalamus, acting as the output to the action selection system, is known to project directly onto local inhibitory neurons in cortex (Swadlow, 2002), enabling direct control of information flow between cortical areas.

Extensions for the Hebb repetition effect. To reproduce the Hebb repetition effect, we extended the model described to this point. Because the Hebb repetition effect requires multiple successive trials, the weights in the $\boldsymbol{M}^{\mathrm{FC}}$ and $\boldsymbol{M}^{\mathrm{CF}}$ association matrices have to have an inter-trial dynamic added. We do this with a slow decay of the weights over time, which also means that previous trials have less influence as time progresses. In addition, another set of associations has to be learned. Either learning direct associations between the Semantic Pointers for positions and the presented list items, or learning forward associations allow the reproduction of the Hebb repetition effect. For the forward associations, associations from $\boldsymbol{f}_i \circledast \boldsymbol{p}_i$ to $\boldsymbol{f}_{i+1}$ are learned. The binding to the position disambiguates these associations between lists, as the cue is only the same if item and position are the same.

The influence of these extensions on the single-trial model are minimal. The new association matrices use a much lower learning rate than $\boldsymbol{M}^{\mathrm{FC}}$ and $\boldsymbol{M}^{\mathrm{CF}}$ and only provide a significant influence after a number of trials. Similarly, the decay has more effect on

longer timescales over multiple trials.

Biological plausibility of the CUE model. The notion of biological plausibility is vague. We have mentioned a variety of ways in which CUE is plausible above, and gather those here to clarify what we mean in this context for this model. We take the model to be plausible in a number of respects, including: respecting qualitative anatomical constraints, using only spike-based computation, and using physiologically determined parameter values.

There are a number of anatomical constraints captured by the CUE model. For instance there is a general mapping of the model to anatomical areas. Succinctly, the TCM-related areas map to the medial temporal lobe (Howard, Fotedar, Datey, & Hasselmo, 2005), with context storage in entorhinal cortex (EC). Given the simplified implementation of the $\boldsymbol{M}^{\mathrm{FC}}$ matrix, and its use of the AML, it maps to the hippocampus. Further evidence for this neuroanatomical mapping can be obtained from the connectivity between hippocampus and EC. The superficial EC provides input to hippocampus, but does not receive direct input for hippocampus (Witter, 2010). In contrast to that, the deep layers of EC receive input from hippocampus and might be relevant for recall, especially the recall of pre-experimental context. This is consistent with the connectivity in the model where the context network projects to the association matrix learning network attributed to hippocampus. The learning network for $\boldsymbol{M}^{\mathrm{FC}}$ also projects back to an ensemble recalling the prior context before it gets combined in a different ensemble. The short-term memory related components of the CUE model can be assumed to correspond to cortical areas, in particular the prefrontal cortex. The prefrontal cortex has been found to be involved in working memory tasks in many studies (e.g., Goldman-Rakic, 1995; Owen, 1997).

A further constraint on this model is that all of the mechanisms are spike-based, including the AML rule. This means that the entire model can be thought of as one large recurrent neural network, with neurons, connection weights, and synapses, and nothing else. All communication between neurons is via spikes sent down axons, which are then

weighted and cause post-synaptic responses. This ensures that the signals being sent throughout the model mimic the variability, frequency content, and dynamics typical of biological brains. As a result, running the CUE model means simulating it for the same lengths of time that are used to run experiments. There is no need to match 'time steps' in the model to seconds in an experiment because the model is run directly in seconds (specifically at a resolution of 1 ms). As a result, high-level (experiment scale) dynamics are fully constrained by low-level (individual neuron) dynamics.

The ways in which these signals are generated and processed giving rise to the observed dynamics are determined by the physiological properties of the cells, both in the brain and in the CUE model. These physiological properties include membrane time constants, synaptic time constants (which vary as a function of the neurotransmitters being used), refractory time constants, and so on. In the CUE model, all of these listed properties are set by using measurements from biological cells. So, for example, feedforward excitatory connections typically use AMPA-type glutamatergic receptors (whereas lateral connections are typically NMDA-type). AMPA receptors have a time constant of about 5ms, whereas NMDA-type have time constants of about 100ms[2]. Incorporating these constraints as fixed parameters, means that the free parameters of the model are small in number, and relate largely to the more cognitive aspects of the model, as discussed below.

In short, the model realizes high-level computations in a neurally plausible (to a degree) substrate. As a result, we may hope to find predictions about neural organization that result from implementing the specific set of computations needed by the model. For instance, circular convolution seems like a particular operator, present throughout the model that may make connectivity predictions. Unfortunately, circular convolution can be implemented by a linear transform followed by a local nonlinear operation. As a result, it's structure is consistent with general features of cortex, up to a certain dimensionality (see Eliasmith (2013), section 4.6). Consequently, the main points of direct comparison between

[2] http://compneuro.uwaterloo.ca/research/constants-constraints.html

the model and single cell data is similarity in physiological responses during task performance, not anatomy. We make such comparisons in the next section.

These features of the model lead us to consider it biologically plausible. Nevertheless, in the next section, we focus on the psychological-level results to demonstrate that these biological details are included in a manner that is consistent with preserving core functions of memory. It should also be noted that, although we have not pursued the possibility here, the low-level properties of the model make it possible to estimate other physiological signals, including fMRI and EEG signals that could be gathered during working memory tasks (see, e.g., Eliasmith (2013), section 5.8).

Results

To validate the CUE model, we matched it to human data from immediate serial recall, immediate free recall, delayed free recall, and continual distractor free recall experiments. The timings of presentation speed, recall durations etc. in the experimental protocols used to obtain the human data, were matched exactly in the model simulations. In addition, we compared the behavior of the AML learning rule to a physiological experiment, in order to demonstrate that it function appropriately independent of this model.

An example of the memory encoding with synaptic weights and neural activities is given in Figure 9. The parameters used to match different experimental conditions are summarized in Table 1. We begin with our characterization of the AML.

The AML accounts for neural changes during association learning

Ison, Quian Quiroga, and Fried (2015) recorded neural data from the medial temporal lobe of 14 human subjects, who had to undergo surgery due to severe epilepsy. They found that during association learning, individual neurons change their firing rapidly to encode newly learned associations. First they identified neurons that responded selectively to the

picture of a certain face or landmark. Then pairs of persons and landmarks were associated and the learning was assessed with multiple tasks while neural recordings were done.

In particular, pair-coding units were identified that showed an elevated firing rate to one of the stimuli, the *preferred* (P) stimulus, before learning (BL) of any associations. After learning (AL), these units would show elevated activity to both the preferred and associated, *non-preferred* (NP) stimulus. The same effect was observed for the normalized population response. Furthermore, no change in this normalized population response was observed for non-associated (NA) stimuli that are neither the preferred nor the non-preferred (but associated) stimulus.

We do not use the CUE model itself to match this data because it currently does not support this sort of experimental task. Instead, we use the simple model shown in Figure 10 to demonstrate that the AML captures long term memory effects independently of the CUE model. The maximum firing rates for the representational space in this model were sampled from $10\,\mathrm{s}^{-1}$ to $20\,\mathrm{s}^{-1}$ and intercepts were uniformly distributed between 0.1 and 1. Each ensemble in the model represents 32-dimensional Semantic Pointers and connections are initialized with an identity transform, but weights change during the simulation according to the AML, between the *pre* and *post* ensemble. To account for neural background firing, low-pass filtered (filter time constant of 0.1 s) Gaussian white noise with a mean of 0.01 and a standard deviation of 0.05 was injected into the neurons. Spikes were recorded from the post population and analyzed as done in Ison et al. (2015).

Figure 11 shows example spike trains of a neuron to the preferred and non-preferred stimulus before and after learning. Both, experimental and model data, show the same qualitative effects. The firing rate increases after stimulus onset for the preferred stimulus. The firing rate for the non-preferred stimulus only increases on stimulus onset after the association has been learned. This change in firing occurs as a result of the AML rule increasing the connection weight strengths between associated stimuli, using its simple Hebbian mechanism (i.e. the rule states that weights will increase if the pre-synaptic vector

increases at the same time as the post-synaptic vector). Critically, the AML is able to learn many such associations without eliminating the firing rate increase of identified neurons, unlike other previously proposed Hebbian rules (e.g. the PES rule; MacNeil and Eliasmith, 2011). This is because the AML uses the presynaptic weight correlations to gate the updates in a weighted fashion.

Similar qualitative matches of the data and model are shown in the population responses in Figure 12. The normalized population activity increases after stimulus onset for the preferred stimulus, and this increase is the same before and after learning. For the non-preferred stimulus, a clear increase in the population activity is only observed after learning. Learning the associations does not influence the population response to non-associated stimuli. The increase in population activity is delayed in the experimental data compared to the model data and decays more quickly. This can be attributed to additional processing or single neuron properties that are not explicitly modeled.

Critically, the AML rule underlies all learning in the CUE model, thus playing a central role in all subsequent results. While spiking data from humans during learning in the tasks simulated is not available to compare to, the model nevertheless simulates this data. As a result, the simulated data is a prediction of the expected spiking dynamics during these tasks, and can be directly compared to such data if it becomes available in the future.

Serial recall

We now consider direct behavioral comparisons of the full CUE model, which includes AML. To test serial recall, ten items were presented at a rate of one item per second. This matches an experimental condition from Jahnke (1968) and allows for a direct comparison given in Figure 13. One minute was provided to recall items, which was sufficient to the model to attempt to recall each of the ten list items. The serial position curve shows the well known primacy and recency effects. The model and experimental data

are statistically similar, as nine of ten confidence intervals overlap.

We also evaluated the transposition errors, which occur rarely in the model, but when they occur, transpositions of nearby positions are more likely. This effect is well known from human experiments (e.g., Henson, 1996), but Jahnke (1968) did not provide any quantitative data that we are able to compare to. Further we looked at the model behaviour when an item was recalled too early. In one instance the model continued with an item later in the list; in six instances the model continued with an item that was omitted earlier in the list; and in 24 instances the model was not able to recall an item at the next list position. This is consistent with human experiments where more fill-in than infill errors are found (Surprenant, Kelley, Farley, & Neath, 2005) and in contrast to item-to-item chaining models that predict a higher frequency of infill errors.

Free recall

Experimental free recall data was taken from experiments 1 and 2 of Howard and Kahana (1999). This same data is modelled in Howard and Kahana (2002) using TCM, which allows direct comparison of our results with those previously published. In these free recall experiments, twelve items were presented with a duration of 1 s (immediate recall) or 1.2 s (delayed and continual distractor recall) each. Besides the immediate recall condition, data was recorded for a delayed and continual distractor condition. In both latter conditions a 16 s distractor task was introduced between the presentation and recall phase. In the continual distractor condition, the same 16 s task was also used in-between every two items. To model the distractor task, non-list items were presented to the model at a rate of ϕ items per second. These items were allowed to influence the short-term memory, but synaptic-weight changes were disabled for these irrelevant items. The duration of the recall phase was 45 s for immediate free recall and 60 s for the remaining conditions.

A total of 100 model instances with different random number seeds was run for each experimental condition, simulating 100 different experimental subjects. That is, we ran 100

subjects worth of lists.

Figures 14 and 15 shows the experimental and model data.

First, the distribution of the total number of successful recalls is shown (top row). To quantify the match of these distributions, we looked at whether the experimentally reported parameters fall within the the 95 % confidence intervals of the model mean, standard deviation, and kurtosis. All experimental parameters lie within the model confidence intervals when comparing the model to the human data, except the standard deviation in the continual distractor recall condition. This indicates that the model approximates the experimental distributions well, even though an equality of the distributions cannot be inferred. Observing the similar shapes in the full histograms of the experimental and model data in the top row of Figure 14 is consistent with this result.

Second, the serial position curves are shown. They display a clear recency effect in the immediate recall condition. It is largely attenuated in the delayed recall condition and partially restored in the continual distractor condition.

Third, the recency effect is also displayed in the probability of first recall. In the immediate recall condition, the last item is recalled first with a high likelihood. Again this is largely attenuated in the delayed recall condition and partially restored in the continual distractor condition.

Fourth, the conditional response probability (CRP) is given, indicating the probability to jump a certain number of position forward or backward between successive recalls. The primary features of the CRP curves are the increase around zero, indicating a preference to recall items together that were presented together, and the asymmetry, indicating a bias to recall items in forward order. These effects are most apparent for immediate recall and get attenuated in delayed and continual distractor recall. For both the model and experimental data, the continual distractor case is more attenuated than the delay case.

We also note that the model is directly capturing individual variability, as

demonstrated by the error bars on the model data. We are essentially simulating 100 individuals to get these results. The variability is a result of random neuron parameter choices, not high-level variable choices (these were set as described previously). However, it is possible to qualitatively describe the effects of changing those high-level parameters. Specifically, we have found that: a) the null choice (aka minimum evidence parameter) μ affects the number of recalled items (the higher μ, the fewer items will be successfully recalled, which has the effect of shifting the serial position curve downwards); b) standard deviation of the input noise σ: affects the total number of recalls and the number of correct recalls (the higher the more recalls are made, but fewer are correct in serial recall due to the position constraint; it also flattens out the CRP); c) the probability to use serial recall (in free recall experiments) ψ boosts the serial recall strategy when increased (this affects the shape of the serial position curve in free recall by raising the recall of the first few items, and P(first recall) also increases for the first position); d) distractor rate ϕ increases the distractor rate which flattens P(first recall) and the serial position curve (this is because the current context when starting recall will be less similar to the context of the last item); e) an increase in the probability of using the serial recall strategy will produce a larger forward asymmetry in the CRP curve.

Overall, the model data matches the experimental data well. Of the 108 free recall data points, 97 of the experimental means fall within the model confidence intervals. That means about 10.2 % of the confidence intervals do not contain the experimental means, while about 5 % are expected to lie out side the intervals by pure chance. However, one deviation of the model data from the experimental data is quite salient: in the delayed free recall condition a higher forward recall bias is predicted by the model than experimentally observed. Interestingly, this feature was the least well fitting aspect of the original TCM model as well.

Neural Representation of Temporal Context

In a recent experiment, Folkerts, Rutishauser, and Howard (2018) recorded spike train data from human subjects during an episodic memory task to test for the existence of a neural representation of temporal context that gradually changes over time. While we have not implemented the same experiment, we expect our neural model of recall to exhibit similar neural responses, as it represents temporal context as well. In the Folkerts et al. (2018) experiment, human subjects were shown 100 images during a study phase. Later during a test phase they were shown 100 images, only 50 of which were from the study phase. They were asked to indicate if they had seen each image before and rate their confidence in that judgement. Microelectrodes were implanted bilaterally in subjects' hippocampi and amygdalae, and a total of 1286 neurons were recorded across 35 subjects and 49 sessions for a total of 10439 trials.

In our model experiment, we use a procedure analogous to a delayed recognition experiment. The model is presented with 10 items at a rate of one item per second. After a delay period of 20 seconds, the 10 items are presented again in random order. This mimics a recognition experiment and the model will perform the same context updates as during presentation, however we do not have the model produce a recognition response or familiarity rating. Including the delay helps to mimic the elapsed time between study and test in the original experiment. We analyzed spikes from 800 (of about 13,000) neurons in the context ensemble which are of primary interest. In addition, we also did the same analysis with neurons providing $\boldsymbol{M}^{\mathrm{CF}}$, $\hat{\boldsymbol{v}}$, $\boldsymbol{p}$, $\hat{\boldsymbol{p}}$, and the memory buffer of the OSE. This means (for each analyzed population) we have more neurons per subject (and analyzed region) than the original experiment, although the original experiment has far more trials (we have 68 subjects with one trial each which is determined by the amount of data we could record before exhausting the memory on our simulation systems). In addition, our recordings are far more targeted than in the original experiment, and thus results are likely to be less noisy.

We adopt our analyses of the spike trains and population vectors from the original work. Spike trains are sliced into "events," corresponding to the presentation of items during the presentation and recall phases. For each event and neuron, the firing rate is averaged and then z-scored across all events considered in the analysis. This results in population vectors (with dimensionality matching the number of recorded neurons) for each trial and event. In our analysis, as in the original, the similarity between pairs of such vectors is considered, normalized by the number of units.

Figure 16 shows the model results. Figure 16a replicates the exponential increase in similarity with increasing recency, shown in Figure 6a in Folkerts et al. (2018), although with less variability. It can also be noted that while the similarity decreases over the timescale relevant to the model, the Folkerts et al. (2018) shows a decline over a much longer time span. We believe this suggests that additional context representations changing on slower time scales, not included in the model, may exist.

It is not possible for us to exactly reproduce Figure 6b from Folkerts et al. (2018), as it relies on confidence judgements. However, Figures 16b and 16d indicate the same effect: after the presentation of an item in the recognition period, the context of the presentation is re-instantiated as seen by the similarity of non-negative lags. In addition, a contiguity effect can be observed with the similarity decreasing as negative lag increases. It is fairly small and non-significant in Figure 16b, but is more apparent when disabling the position input in the model (Fig. 16d). In the recognition experiment, the position only provides additional noise in the context representation as the position during the recognition phase is unknown. In general, we expect this effect in direction of negative lags to be less strong due to the way old context is added into the context representation: it is only partially present in the context that gets associated to an item and during re-instantiation it gets further attenuated by being added into the current context. Compare this to Fig. 4, summing both curves gives the shape expected in Fig. 16b/d.

Note that negative similarity values in Figure 16 are are caused by lower mean firing

rates in the recognition phase as we are inhibiting the position input (the correct position is not known in the recognition phase), resulting in less overall input to context neurons.

Applying the analysis to the output spikes of $\boldsymbol{M}^{\mathrm{CF}}$ yields the same, but slightly noisier, results. This is expected because this neuron population essentially applies a linear transform. With the exception of the OSE spikes, all other analyzed populations did not produce any contiguity effect or other similarity relationship in dependence of the lag. Figure 16c shows that in the OSE population similarity is low in general, but increases as the lag increases. This could be caused by the delay period being too short to fully decorrelate the firing of the OSE neurons from the presentation phase and higher lags will be more likely to be closer in time to the recognition period.

Hebb repetition effect

The extended CUE model can account for the Hebb repetition effect. We compare to the data from Hebb (1961) and again reproduce the experimental protocol as closely as possible. Given that there were the 25 experimental subjects, a total of 25 model instances were run for 24 consecutive trials each with nine item lists (digits from one to nine in random order, without repetitions). Every third list (starting with the third trial) was repeated.

Two additional parameters are introduced through the extension for the Hebb repetition effect. The learning rate for the direct position to item associations or forward associations is set to 0.05 or 0.25 respectively. The decay rate for the $\boldsymbol{M}^{\mathrm{FC}}$ and $\boldsymbol{M}^{\mathrm{CF}}$ associations was set to decay to 0.2 of the original weight within a minute. Recall that this parameter captures inter-trial dynamics, which are not included in the original model.

The general recall performance of the model is slightly worse than observed in the human data (Figure 17). Nevertheless, the mean number of correct recall increases by two to three items for those repeated over the course of the experiment, which matches the experimental data. Thus, the Hebb repetition effect is reproduced.

In Figure 18, we demonstrate that the model can learn multiple repeated sequences, while performing at baseline for non-repeated sequences. In this simulation, both the second and third list were repeated with the first, fourth, seventh list and so on being random. For both repeated lists, a Hebb effect or improved recall was observerd. This demonstrates that the position representations are not being confused across lists, a concern for some models of the Hebb effect (Burgess & Hitch, 2006).

More subtly, in Figure 19, we show that the model does not exhibit a Hebb effect in position-item associations, but does in forward associations for a list with the first four items repeated and the remaining items being randomized. Using forward associations, the recall performance increases for the first four non-random items, but it does not using position-item associations. In human subjects, changing some of the items in the list generally does not produce a Hebb effect (Burgess & Hitch, 2006), suggesting that position-item learning may be a more appropriate model for human subjects.

Discussion

We have presented what we believe is the first spiking neural network model of combined activity-based short-term memory and weight-based long-term memory. The same model matches behavioral data from serial and free recall experiments, but in contrast to previous models provides a neural, mechanistic characterization of memory function. Perhaps most uniquely, the model also implements the necessary control mechanisms for switching between encoding and recall under a variety of task conditions.

Core behavioral findings in memory research are reproduced by the CUE model. In serial and immediate free recall, the CUE model shows the primacy and recency effect. The recency effect is attenuated in the delayed free recall condition, but reappears in continual distractor free recall. Transposition errors are a frequent error type, but uncommon in comparison to the number of items recalled in the correct position in serial recall. Transpositions of nearby items are more likely than transpositions of items further apart.

Furthermore, in (immediate) free recall the model is most likely to begin recall at the end of the list, recall nearby items together, and is biased towards forward recall. We also note that several effects not demonstrated explicitly here are likely captured by CUE because it is combining aspects of the OSE and TCM models. For instance, the OSE has been shown to capture immediate, delayed, confusable item, and fill-in effects in serial working memory (Choo, 2010), which are expected to be transferred to CUE. All of these observations are in accordance with experimental data.

The CUE model not only produces these qualitative effects, but also provides good quantitative matches to experimental data (Figs. 13, 14). Only the CRP curve for the delayed free recall condition shows a clear deviation from the data by predicting a too strong forward bias. This particular aspect happens to be also the least well matched in the original TCM (Howard & Kahana, 2002). In a more recent version of the TCM, the same prediction is even closer to the prediction of the CUE model (Sederberg et al., 2008). This may indicate that both the TCM and CUE models are not capturing an essential aspect of memory, potentially related to how the context signal evolves under this particular experimental condition. As a result, this remains an area of future work.

Apart from this deviation, the CUE model produces slightly less of a contiguity effect than found experimentally in the continual distractor free recall condition. This could potentially be caused by the reduced associative strengths through the distractor intervals that is more susceptible to the neural noise in the simulation. In the CUE model the retrieval depends on the absolute activation, whereas in the Luce choice rule in the TCM model depended only on the relative activation.

The TCM model has previously been criticized for being unable to simultaneously match the probability of first recall and CRP curves (Farrell & Lewandowsky, 2008; but see Howard, Sederberg, & Kahana, 2009). We highlight that the CUE model simultaneously matches both data sets. It also addresses another criticism expressed by (Farrell & Lewandowsky, 2008) by using the contextual evolution during the recall process.

When the CUE model is extended with slow learning of either direct position to item associations or forward associations, the Hebb repetition effect can be reproduced qualitatively. Because we were unsuccessful in obtaining the effect without this extension, we predict that long-term memory might employ multiple encoding schemes. One scheme associates the context signal to items via one-shot learning, while another scheme requires multiple learning trials and encodes position-item or forward associations.

We also note that this model, unlike many, is able to account for both average performance and individual variability. For instance, in the top row of figure 14, we show that the distribution of individual performance matches between the model and human subjects, as well as the mean. In both cases, this match suggests that samples are taken from a similar underlying distribution. Notably, the reason for the variation in the model is because many of the neuron parameters (e.g. gains, biases, encoders) are randomly selected from distributions that match low-level neural parameter distributions found in cortex (Eliasmith & Anderson, 2003). Consequently, the observed behavioral variability is a direct consequence of neural variability because high-level behavioral parameters are kept constant; suggesting that the proposed abstract model captures core features of the underlying biological networks. Many existing models would require a modification of the high-level behavioral parameters to achieve such variability which does not provide an explanation of what causes the variability in these parameters.

Providing such connections between behavioral level observations and neural level mechanisms is perhaps the most unique aspect of the CUE model. Constructing a model that satisfies low-level neural constraints, while performing the appropriate computations to capture cognitive behavior provides an important "linking of levels" to improve our understanding of brain function. We note that the CUE model not only shows that the necessary computations can be performed in spiking neurons, but also that neural components can be arranged and externally controlled to perform all the stages of these memory tasks without 'rewiring' the model on-the-fly. Consequently, some of the

constraints on information flow, noise, and dynamics that must be respected by the brain are incorporated into the CUE model, in contrast to past models. We believe that this may provide a test bed for exploring what happens when aspects of these constraints are disrupted or changed through external interventions (e.g. drugs, stimulation or ablation).

More specifically, we have also compared mechanisms in the CUE model to neural recordings. Such comparisons are rare because neural recordings can be obtained ethically from humans only in very specific circumstances. While the CUE model currently does not implement learning of associations between items, we showed that the underlying learning rule is able to account for neural changes during such learning observed in humans. As well, the model generates neural representations of temporal context that match available single physiological data from humans on context reinstantiation.

The spiking neural implementation also helps constrain many parameters values, such as synaptic and membrane time constants. These have been set to biologically plausible values derived from experimental findings[3] in the CUE model, and have not been adjusted to achieve the parameter matches. Along the same lines, it is worth noting that the connection weights are fixed (where the AML is not employed) and determined by least-squares optimization for specific functions prior to the simulation. As a result, the model has few *free* parameters (Table 1), despite having millions of parameters (neural connection weights, time constants, and so on).

In particular, we note that the learning rate and distractor rate do not change for any simulations. The probability of using a serial recall strategy is 1 for serial recall and 0.1 for free recall, an expected variation driven by task demands. Interestingly, the noise parameter also varies between free and serial recall, which is perhaps a result of a less structured recall process in free recall. The one exception is the continual distractor case for which the serial recall value gave a noticably improved fit. Finally, μ, the bias of the null choice (or minimum evidence) parameter, which acts as a kind of threshold, changes

[3] http://compneuro.uwaterloo.ca/research/constants-constraints.html

between experiments. At first it is not clear why that threshold should change. Upon further reflection, however, there is a systemmatic nature to the effective parameter values. Namely, the threshold decreases as the distractor difficulty of the task increases. As a result, the model suggests that with increased distractors subjects allow less certain matches to be reported (lowering the minimum evidence means allowing less good matches to pass the threshold). Overall, no more than four parameters have been adjusted for all of the data matches described. Two additional parameters (weight decay rate and learning rate for the position-item or forward association) need to be added for the extensions to the Hebb repetition effect. Note that the OSE and TCM parameters $\gamma = 0.99775$ and $\beta = 0.62676$ were set from previously published values and not further adjusted to match the experimental data.

To match the experimental data for the different task conditions, it was necessary to adjust some of the free parameters. The remaining differences in parameter values relate to the recall networks, and to the influence of extra-list items not explicitly modeled that might differ between task conditions. In particular, the minimum signal strength decreased from 0.04 to 0.03 with increasing task difficulty due to longer distractor phases (i.e., the recall needs to be more sensitive when items have more time to decay in memory). Less clear is the reason for the difference of standard deviation σ of the noise on the recall input. Note that no exhaustive search of parameter values has been performed due to prohibitively long simulation times of a model of this size. Thus, one can expect some robustness against deviation from the parameter values as it would be unlikely to hit good parameters very precisely without an exhaustive search if the model is not robust.

Notably, despite building and simulating the model in neurons, it is still possible to characterize qualitative performance changes in terms of only a few parameters. Several examples of these are discussed in the Free Recall section above. In addition to those, we note that: a) transposition errors are increased by more noise σ; b) recency can be controlled by adjusting the connection strength from STM/OSE into the recall network,

where reducing the connection strength reduces recency; c) the TCM β parameter has an influence on recency (and primacy), with a higher value leading to less recency and primacy (and a lower value leading to more primacy and less recency in serial recall, but more recency in free recall); d) for contiguity we observe that a smaller TCM β value increases recall and keeps the CRP asymmetric, but increases the probability of longer lags; and e) symmetry in the CRP curve increases as σ is decreased or μ is increased.

While the CUE model is based on prior models of memory, it improves on these and extends them. More detail is added to the OSE model by replacing the episodic memory trace with a plausible weight-based mechanism, and introducing a neural implementation to provide the position Semantic Pointers. With regard to the TCM, the CUE model demonstrates its biological plausibility given some critical adjustments. Furthermore, it demonstrates that the TCM can be plausibly considered to be part of a dual-store model, despite being initially proposed as a single-store model. As well, the CUE model makes the recall process part of the model and implements it in spiking neurons. In many prior models of memory (e.g. Brown et al., 2000) the recall process is not treated systematically, or included as part of the model. Finally, the CUE model can be extended to multi-trial experiments to investigate effects like the Hebb repetition effect, while most existing memory models focus on single-trial experiments.

While the CUE model provides largely qualitative matches to neural data and manipulations, we believe it is a good example of how to provide a close tie between neural mechanisms and established behavioral effects. To this end, it provides a clear example of how neural and behavioral data can provide complementary constraints on a single underlying model. In future work, we expect to increase both the neural fidelity and the breadth of behavioral effects reproducible within the model. However, we believe that even at this early stage, the model demonstrates the utility of simultaneously considering psychological and neuroscientific targets of explanation at the same time. For example, by means of the AML, the CUE model matches behavioral results from memory experiments

while at the same time providing a possible neural mechanism reproducing neural data.

References

Abraham, W. C. & Robins, A. (2005). Memory retention – the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, *28*(2), 73–78. doi:10.1016/j.tins.2004.12.003

Anderson, J. A. (1973). A theory for the recognition of items from short memorized lists. *Psychological Review*, *80*(6), 417–438. doi:10.1037/h0035486

Baddeley, A. D. (1986). *Working memory*. Oxford Psychology Series. Oxford, England: Clarendon Press.

Baddeley, A. D. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences*, *4*(11), 417–423. doi:10.1016/S1364-6613(00)01538-2

Bekolay, T., Kolbeck, C., & Eliasmith, C. (2013). Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks. In *35th Annual Conference of the Cognitive Science Society* (pp. 169–174). Cognitive Science Society.

Botvinick, M. M. & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, *113*(2), 201–233. doi:10.1037/0033-295X.113.2.201

Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, *107*(1), 127–181. doi:10.1037/0033-295X.107.1.127

Burgess, N. & Hitch, G. J. (1992). Toward a network model of the articulatory loop. *Journal of Memory and Language*, *31*(4), 429–460. doi:10.1016/0749-596X(92)90022-P

Burgess, N. & Hitch, G. J. (2006). A revised model of short-term memory and long-term learning of verbal sequences. *Journal of Memory and Language*. Special Issue on Memory Models, *55*(4), 627–652. doi:10.1016/j.jml.2006.08.005

Choo, X. (2010). *The ordinal serial encoding model: Serial memory in spiking neurons* (Master Thesis, University of Waterloo, Waterloo, Ontario, Canada). Retrieved from https://uwspace.uwaterloo.ca/handle/10012/5385

Davelaar, E. J., Usher, M., Haarmann, H. J., & Goshen-Gottstein, Y. (2008). Postscript: Through TCM, STM shines bright. *Psychological Review*, *115*(4), 1116–1118. doi:10.1037/0033-295X.115.4.1116

Eichenbaum, H. (2001). The hippocampus and declarative memory: Cognitive mechanisms and neural codes. *Behavioural Brain Research*, *127*(1-2), 199–207. doi:10.1016/S0166-4328(01)00365-5

Eliasmith, C. (2005). A unified approach to building and controlling spiking attractor networks. *Neural computation*, *7*(6), 1276–1314.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition.* New York, NY: Oxford University Press.

Eliasmith, C. & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems.* Cambridge, MA: MIT Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, *338*(6111), 1202–1205. doi:10.1126/science.1225266

Estes, W. K. (1955). Statistical theory of spontaneous recovery and regression. *Psychological Review*, *62*(3), 145–154. doi:10.1037/h0048509

Estes, W. K. (1972). An associative basis for coding and organiation in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). The Experimental Psychology Series. Washington, D.C.: V. H. Winston & Sons.

Farrell, S. & Lewandowsky, S. (2008). Empirical and theoretical limits on lag recency in free recall. *Psychonomic Bulletin & Review*, *15*(6), 1236–1250. doi:10.3758/PBR.15.6.1236

Folkerts, S., Rutishauser, U., & Howard, M. W. (2018). Human episodic memory retrieval is accompanied by a neural contiguity effect. *Journal of Neuroscience*, *38*(17), 4200–4211. doi:10.1523/JNEUROSCI.2312-17.2018

Gayler, R. W. (2004). Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience. Retrieved March 19, 2018, from http://arxiv.org/abs/cs/0412059

Goldman-Rakic, P. S. (1995). Cellular Basis of Working Memory. *Neuron*, *14*(3), 477–485. doi:10.1016/0896-6273(95)90304-6

Gosmann, J. (2018). *An Integrated Model of Contex, Short-Term, and Long-Term Memory* (PhD Thesis, University of Waterloo, Waterloo, ON).

Gosmann, J. & Eliasmith, C. (2015). A Spiking Neural Model of the n-Back Task. In *37th Annual Meeting of the Cognitive Science Society* (pp. 812–817).

Gosmann, J. & Eliasmith, C. (2017). Automatic Optimization of the Computation Graph in the Nengo Neural Network Simulator. *Frontiers in Neuroinformatics*, *11*. doi:10.3389/fninf.2017.00033

Gosmann, J., Voelker, A. R., & Eliasmith, C. (2017). A Spiking Independent Accumulator Model for Winner-Take-All Computation. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*. CogSci 2017. Austin, TX: Cognitive Science Society.

Hasselmo, M. E. (2012). *How we remember: Brain mechanisms of episodic memory*. Cambridge, MA: MIT Press.

Hebb, D. O. (1961). Distinctive features of learning in the higher animal. In J. F. Delafresnaye (Ed.), *Brain mechanisms and learning* (pp. 37–46). Oxford, England: Blackwell.

Henson, R. N. A. (1996). *Short-term memory for serial order* (Dissertation, University of Cambridge, Cambridge, England). unpublished.

Hintzman, D. L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review*, *95*(4), 528–551. doi:10.1037/0033-295X.95.4.528

Hoshi, E. (2006). Functional specialization within the dorsolateral prefrontal cortex: A review of anatomical and physiological studies of non-human primates. *Neuroscience Research*, *54*(2), 73–84.

Howard, M. W., Fotedar, M. S., Datey, A. V., & Hasselmo, M. E. (2005). The temporal context model in spatial navigation and relational learning: Toward a common explanation of medial temporal lobe function across domains. *Psychological Review*, *112*(1), 75–116. doi:10.1037/0033-295X.112.1.75

Howard, M. W. & Kahana, M. J. (1999). Contextual variability and serial position effects in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(4), 923–941. doi:10.1037/0278-7393.25.4.923

Howard, M. W. & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, *46*(3), 269–299.

Howard, M. W., Sederberg, P. B., & Kahana, M. J. (2009). Reply to Farrell and Lewandowsky: Recency-contiguity interactions predicted by the temporal context model. *Psychonomic Bulletin & Review*, *16*(5), 973–984.

Hua, S. E., Houk, J. C., & Mussa-Ivaldi, F. A. (1999). Emergence of symmetric, modular, and reciprocal connections in recurrent networks with Hebbian learning. *Biological Cybernetics*, *81*(3), 211–225.

Ison, M. J., Quian Quiroga, R., & Fried, I. (2015). Rapid Encoding of New Memories by Individual Neurons in the Human Brain. *Neuron*, *87*(1), 220–230. doi:10.1016/j.neuron.2015.06.016

Jahnke, J. C. (1968). Delayed recall and the serial-position effect of short-term memory. *Journal of Experimental Psychology*, *76*(4), 618–622.

Levy, W. B., Hocking, A. B., & Wu, X. (2005). Interpreting hippocampal function as recoding and forecasting. *Neural Networks*. Computational Theories of the Functions of the Hippocampus, *18*(9), 1242–1264. doi:10.1016/j.neunet.2005.08.005

Ma, L., Steinberg, J. L., Hasan, K. M., Narayana, P. A., Kramer, L. A., & Moeller, F. G. (2011). Working memory load modulation of parieto-frontal connections: Evidence from dynamics causal modeling. *Human Brain Mapping*, *33*(8), 1850–1867.

MacNeil, D. & Eliasmith, C. (2011). Fine-tuning and the stability of recurrent neural networks. *PLoS ONE*, *6*.

Milford, M., Wyeth, G., & Prasser, D. (2004). RatSLAM: A hippocampal model for simultaneous localization and mapping. (Vol. 1, pp. 403–408). IEEE international conference on robotics and automation. doi:10.1109/ROBOT.2004.1307183

Murdock, B. B. (1993). TODAM2: A model for the storage and retrieval of item, associative, and serial-order information. *Psychological Review*, *100*(2), 183–203. doi:10.1037/0033-295X.100.2.183

Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (TODAM2). *Psychological Review*, *104*(4), 839–862. doi:10.1037/0033-295X.104.4.839

Norman, K. A. & O'Reilly, R. C. (2003). Modeling hippocampal and neocortical contributions to recognition memory: A complementary-learning-systems approach. *Psychological Review*, *110*(4), 611–646. doi:10.1037/0033-295X.110.4.611

Oliver Trujillo. (2014). *A spiking neural model of episodic memory encoding and replay in hippocampus* (Master Thesis, University of Waterloo, Waterloo, Ontario, Canada).

Owen, A. M. (1997). The Functional Organization of Working Memory Processes Within Human Lateral Frontal Cortex: The Contribution of Functional Neuroimaging. *European Journal of Neuroscience*, *9*(7), 1329–1339. doi:10.1111/j.1460-9568.1997.tb01487.x

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*(3), 623–641.

Raaijmakers, J. G. & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, *88*(2), 93–134. doi:10.1037/0033-295X.88.2.93

Rolls, E. T. (2013). The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in Systems Neuroscience*, *7*(74). doi:10.3389/fnsys.2013.00074

Sederberg, P. B., Gershman, S. J., Polyn, S. M., & Norman, K. A. (2011). Human memory reconsolidation can be explained using the temporal context model. *Psychonomic bulletin & review*, *18*(3), 455–68. doi:10.3758/s13423-011-0086-9

Sederberg, P. B., Howard, M. W., & Kahana, M. J. (2008). A context-based theory of recency and contiguity in free recall. *Psychological Review*, *115*(4), 893–912. doi:10.1037/a0013396

Shiffrin, R. M. & Steyvers, M. (1997). A model for recognition memory: REM—retrieving effectively from memory. *Psychonomic Bulletin & Review*, *4*(2), 145–166. doi:10.3758/BF03209391

Surprenant, A. M., Kelley, M. R., Farley, L. A., & Neath, I. (2005). Fill-in and infill errors in order memory. *Memory*, *13*(3-4), 267–273.

Swadlow, H. A. (2002). Thalamocortical control of feed–forward inhibition in awake somatosensory 'barrel' cortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *357*(1428), 1717–1727. doi:10.1098/rstb.2002.1156

Talmi, D., Grady, C. L., Goshen-Gottstein, Y., & Moscovitch, M. (2005). Neuroimaging the serial position curve. A test of single-store versus dual-store models. *Psychological Science*, *16*(9), 716–723. doi:10.1111/j.1467-9280.2005.01601.x

Usher, M. & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review*, *108*(3), 550–592. doi:10.1037/0033-295X.108.3.550

Witter, M. P. (2010). Connectivity of the Hippocampus. In V. Cutsuridis, B. Graham, S. Cobb, & I. Vida (Eds.), *Hippocampal Microcircuits* (5, pp. 5–26). Springer Series in Computational Neuroscience. Springer New York. doi:10.1007/978-1-4419-0996-1_1

Yu, Q., Tang, H., Hu, J., & Tan, K. C. (2017). A Hierarchically Organized Memory Model with Temporal Population Coding. In *Neuromorphic Cognitive Systems* (126, pp. 131–152). Intelligent Systems Reference Library. Springer International Publishing. doi:10.1007/978-3-319-55310-8_7

Table 1

Summary of free parameters values for distractor rate ϕ, probability ψ of using the serial recall strategy, bias of the null choice μ in recall, standard deviation of the input noise σ in recall, and the AML learning rate η for $\boldsymbol{M}^{\mathrm{CF}}$ and $\boldsymbol{M}^{\mathrm{FC}}$. See text for discussion of the parameter choices and two additional parameters in the Hebb repetition condition not listed in the table.

Experimental condition	ϕ/s^{-1}	ψ	μ	σ	η
Immediate serial recall	—	1	0.0375	0.009	10
Free recall					
Immediate	—	0.1	0.04	0.015	10
Delayed	0.35	0.1	0.0325	0.015	10
Continual distractor	0.35	0.1	0.03	0.009	10
Hebb repetition	—	1	0.015	0.009	10

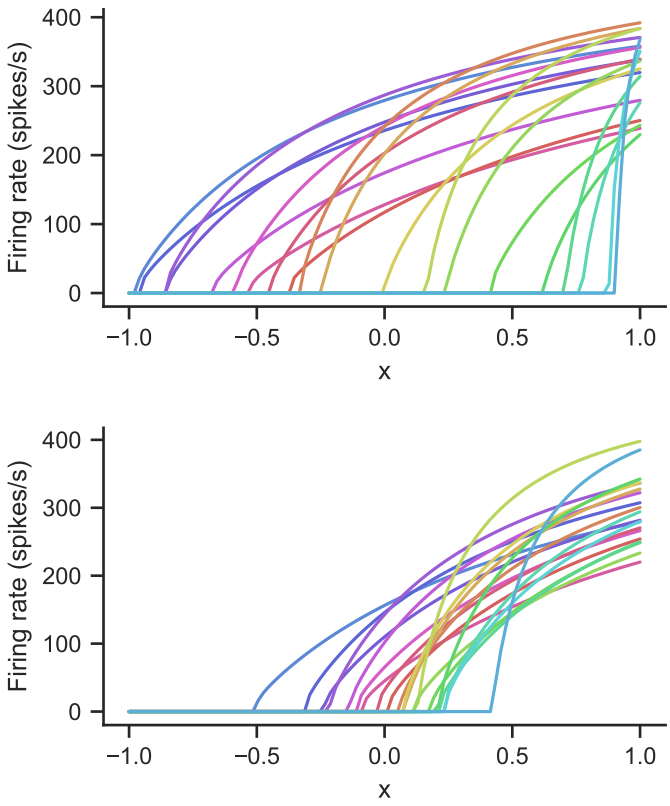


Figure 1. (Top) A population of 20 neurons plotted along their preferred direction vector with x-intercepts and maximum firing rates chosen from an even distribution. These curves show the expected spike rates if inputs are held constant. (Bottom) A population of 20 neurons plotted along their preferred direction vector with x-intercepts chosen from the proposed distribution (Equation 2) for d=32 and maximum firing rates chosen from an even distribution.

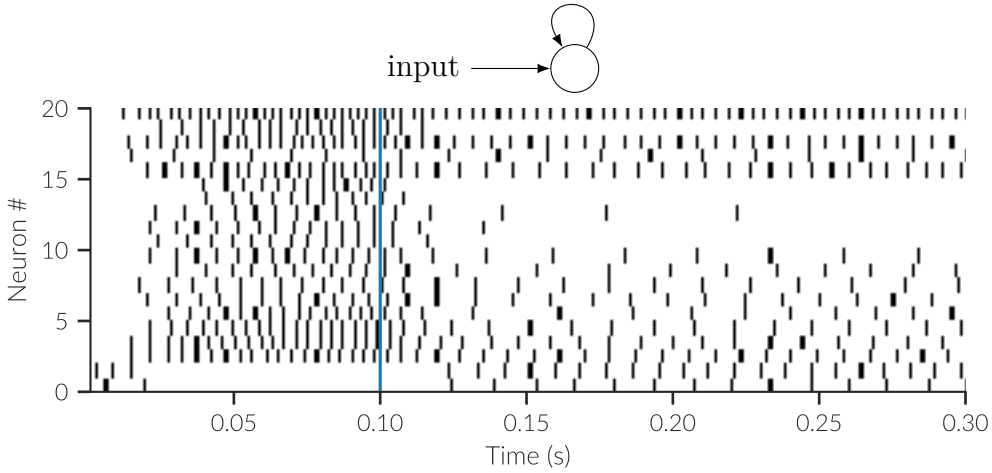
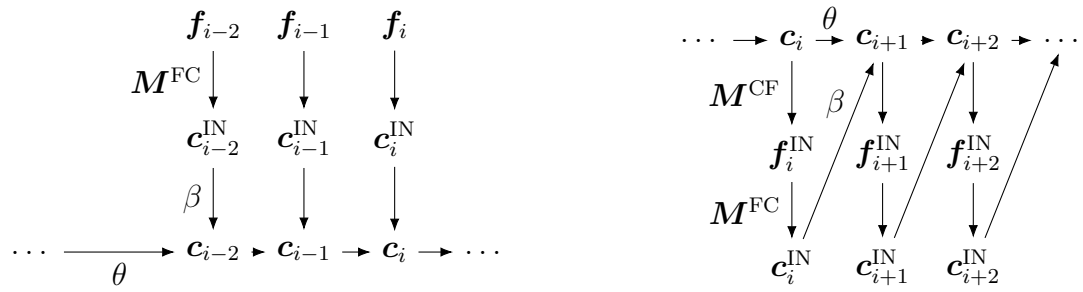


Figure 2. (Top) The simple network structure of the item memory. (Bottom) Spiking activity of 20 example neurons during and after item presentation. Note the sustained firing in the neurons during the memory period (with no item being shown), as observed in cortex. The model has 1000 neurons representing a 32-dimensional item input using LIF neurons, with an input time constant of 10ms and a recurrent synaptic time constant of 100ms.



(a) (b)

Figure 3. Evolution of the context in the TCM during (a) item presentation and (b) recall.

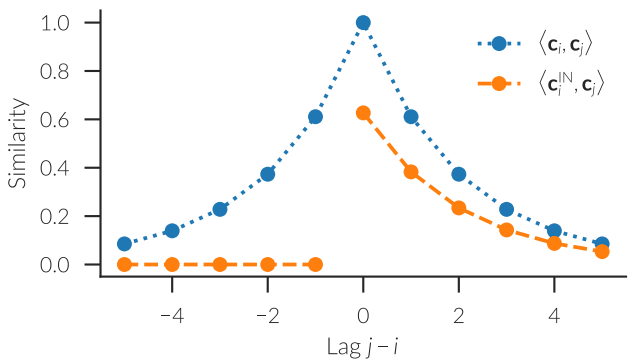


Figure 4. Similarity of the context to itself ($\boldsymbol{c}_i\boldsymbol{c}_j$) and to the retrieved context ($\boldsymbol{c}_i^{\text{IN}}\boldsymbol{c}_j$) for different lags $j - i$.

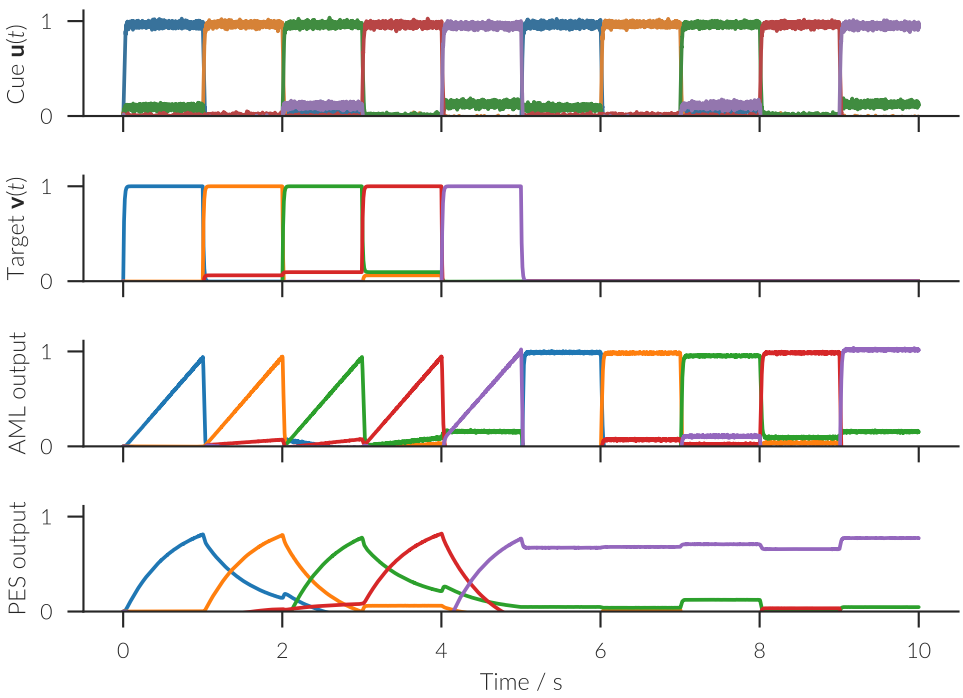


Figure 5. Learning and recall testing of five cue-target pairs with the AML and PES. Each colored trace is the dot product with one of the vectors used. The cue vectors $\boldsymbol{u}(t)$ and target vectors $\boldsymbol{v}(t)$ are distinct, randomly chosen vectors. Both rules have the same cues and targets to learn. Before 5s, the networks are trained using the learning rule listed on the y-axis. After 5 s the cues are shown and the network should generate the associated target. The AML network produces highly similar vectors to the original targets given the cues. The PES network produces the last shown target for all cues.

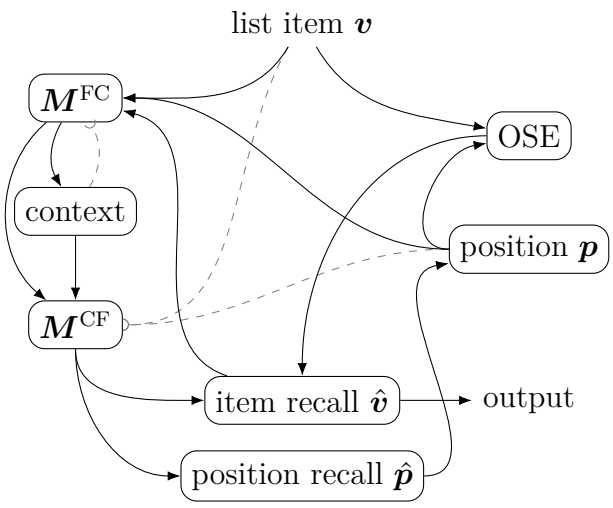
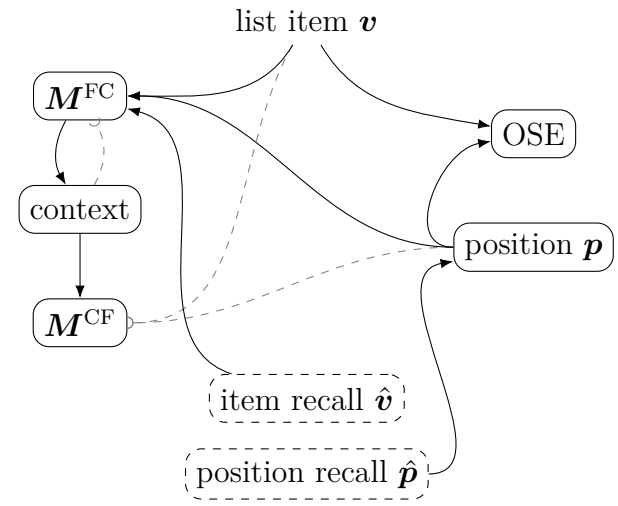
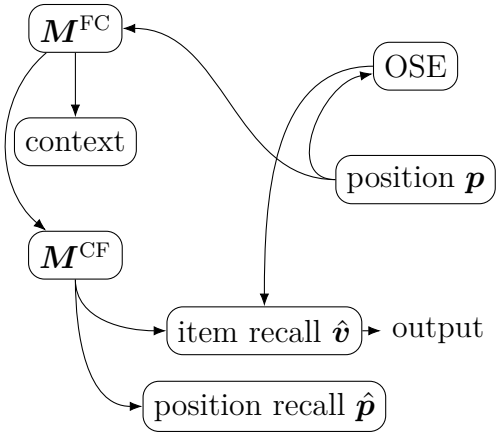


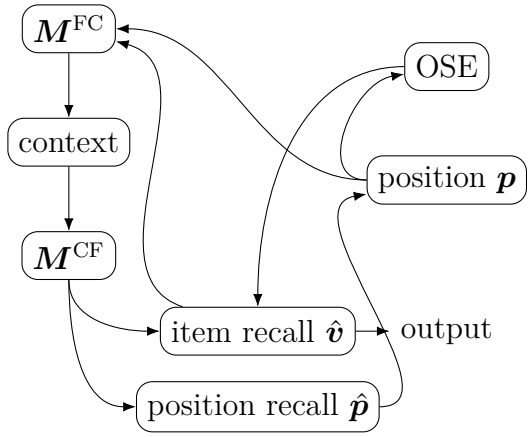
Figure 6. General information flow in the CUE model. The routing of information depending on the task, and task phase is not shown in this figures. Thus, not all shown connections are active at all times.



(a) Presentation phase



(b) Serial recall



(c) Free recall

Figure 7. Information flow during the presentation (a) and recall phases (b, c).

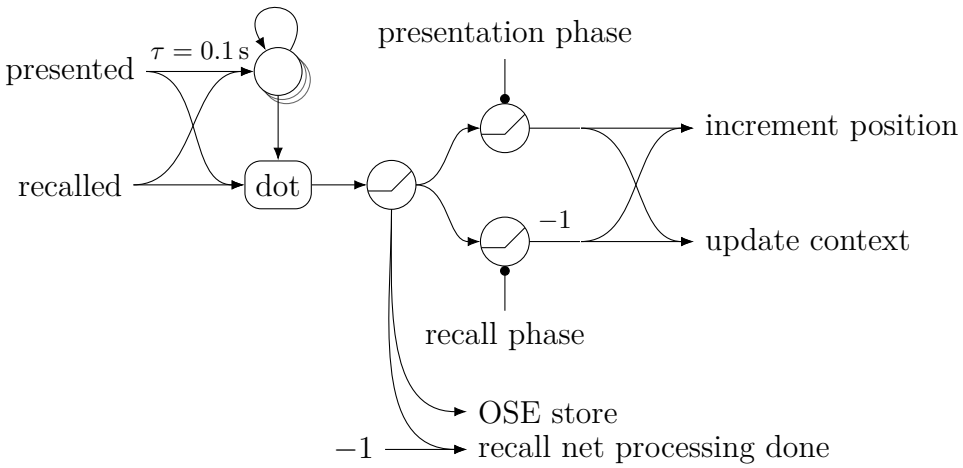


Figure 8. Generation of control signals from the currently presented or recalled item. Neural ensembles are represented with circles. If the circle shows a rectified function, the neuron parameters have been specifically chosen to rectify the input value. The box labeled *dot* is a network computing a dot product.

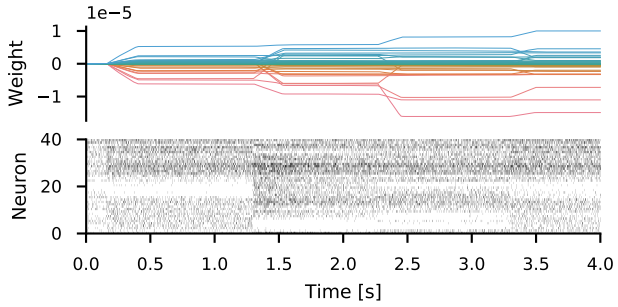


Figure 9. Encoding of four list items with synaptic weights (top) and neural activity (bottom) in the CUE model.

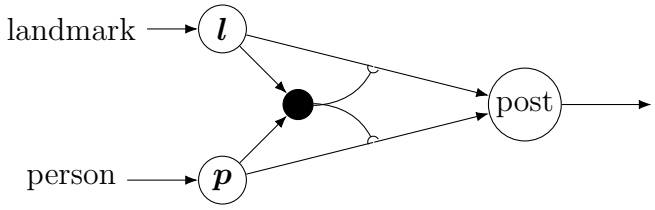
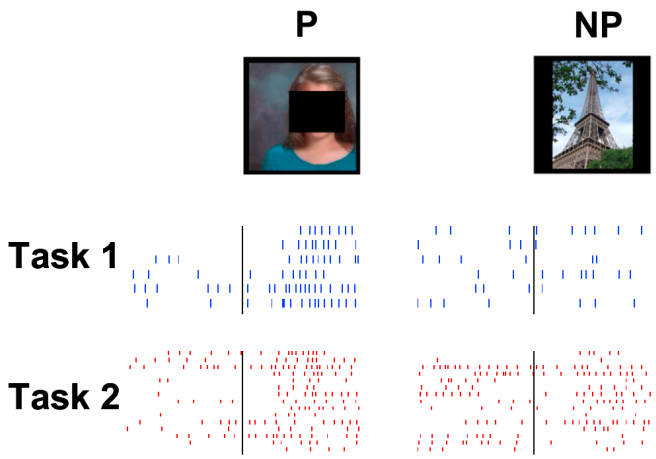
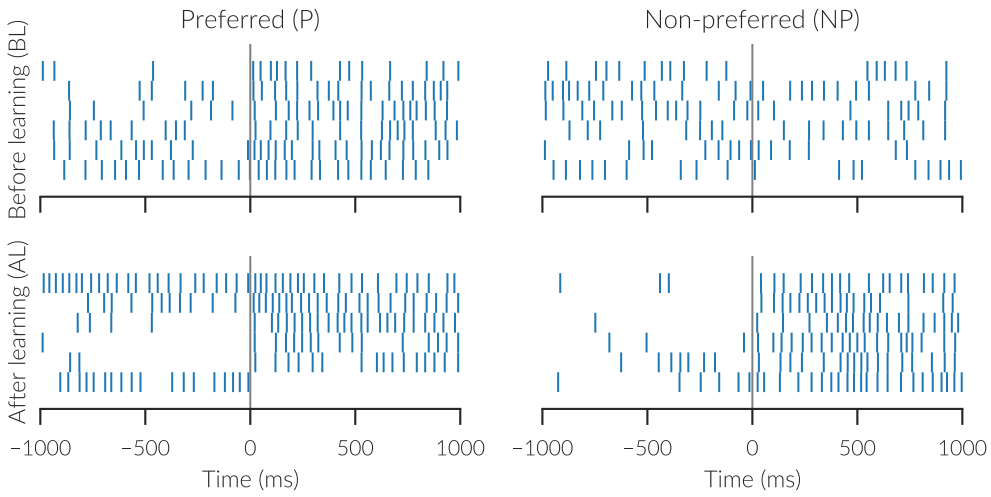


Figure 10. NEF network for associating two stimuli (persons with landmarks here) with the AML. Neural ensembles are represented with labeled circles. The black circle provides the additive combination of its inputs and is not neurally implemented, but equivalent to separate connections. It is used here to simplify the network structure without impacting biological plausibility. Connections ending in a half-circle are modulatory signals to adjust the targeted connections with the AML.

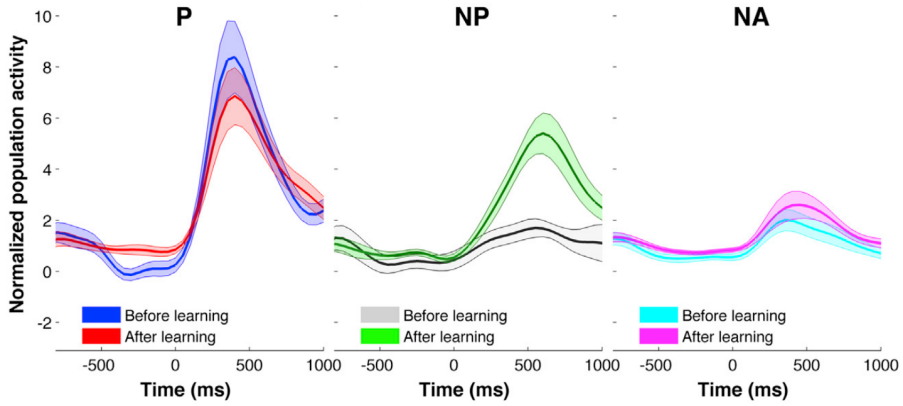


(a) Experimental data

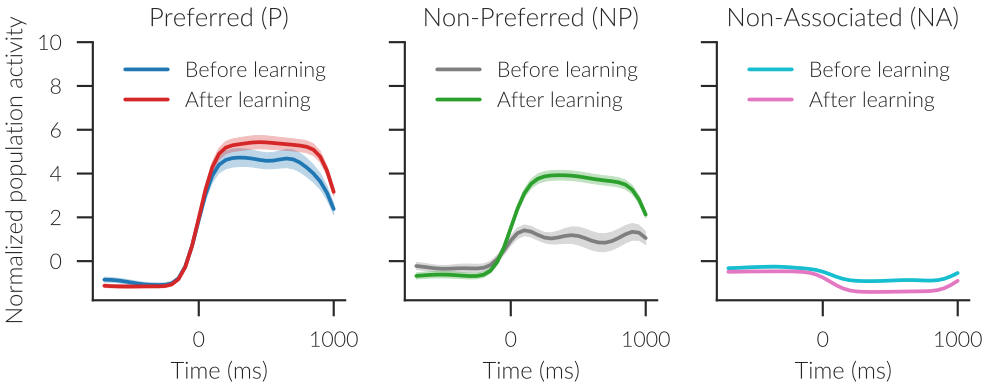


(b) Model data

Figure 11. Change in spiking behavior when learning associations. (a) Exemplary spikes recorded from human hippocampus for the preferred (P) and non-preferred (NP) stimulus. Task 1 is before, task 2 after learning the P-NP association. The black line marks the stimulus onset. Figure adopted from Ison, Quian Quiroga, and Fried (2015) under the Creative Commons Attribution 4.0 International license. (b) Spikes recorded from the NEF model learning the P-NP association with the AML. Both model and experimental data increase in density for both BL and AL for P neurons, have no change after the stimulus BL for NP neurons, and have increased activity after the stimulus AL for NP neurons.



(a) Experimental data



(b) Model data

Figure 12. Population response for pair-coding units to the preferred (P), non-preferred (NP), and non-associated (NA) stimuli before and after learning. Times are relative to stimulus onset. Shaded regions indicate the standard error of mean. (a) Normalized population activity from (a) experimental and (b) model data. Model and experimental data both remain similar and high for P neurons, diverge for NP neurons and remain similar and low for NA neurons. Figure (a) adopted from Ison, Quian Quiroga, and Fried (2015) under the Creative Commons Attribution 4.0 International license.

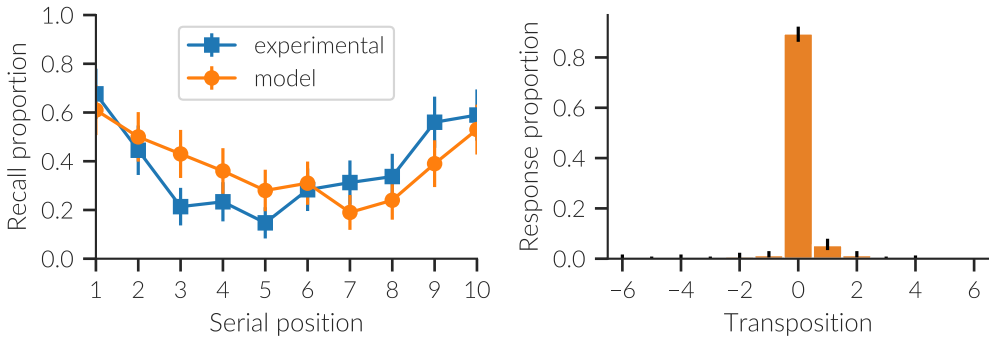


Figure 13. Serial position curve (left) and transpositions (right) for the serial recall of a 10 item list with the CUE model. The experimental data from Jahnke (1968) is shown for comparison in the serial position curve (blue squares). Error bars represent 95 % confidence intervals.

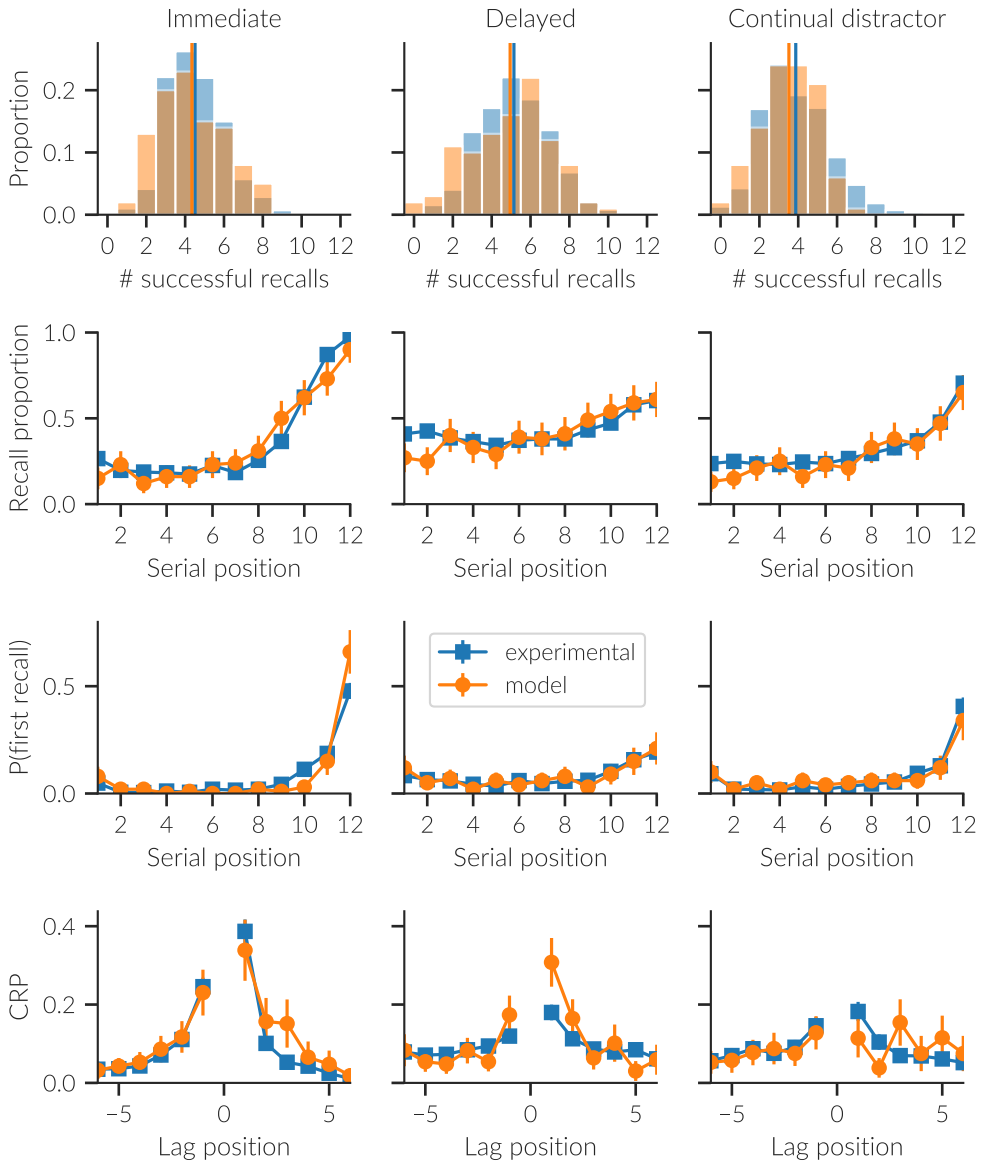


Figure 14. Top row: distribution of total number of recalled items. The distribution means are marked by vertical lines. Remaining rows from top to bottom: serial position curve, probability of first recall, and conditional response probability (CRP). The columns show data from immediate, delayed, and continual distractor free recall. Data from the CUE model is shown in orange, whereas experimental data from Howard and Kahana (1999) is shown in blue. Error bars represent 95 % confidence intervals

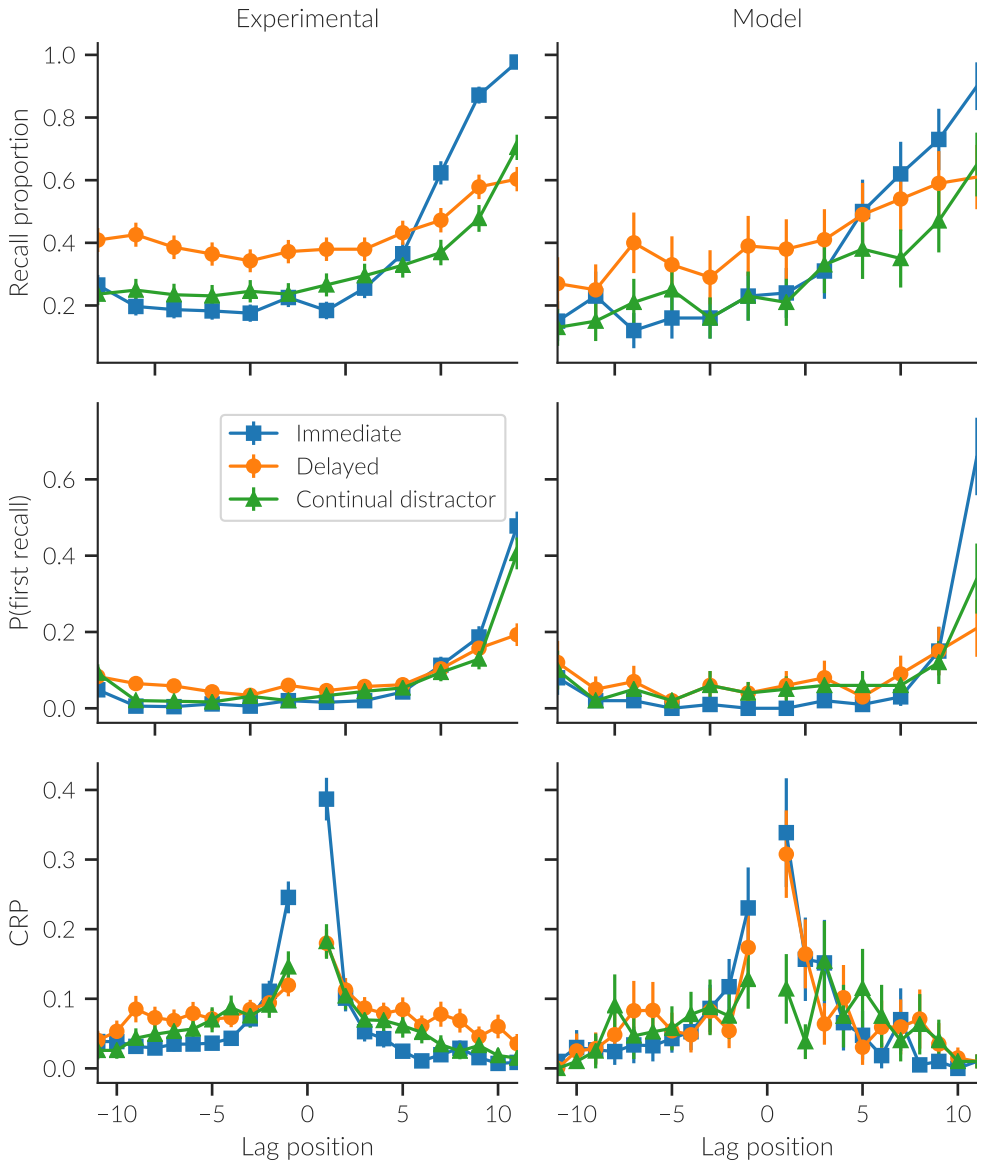


Figure 15. Each of the same types of plots as shown in the rows in Figure 14 but overlaying the model data and original data to aid comparison across conditions.

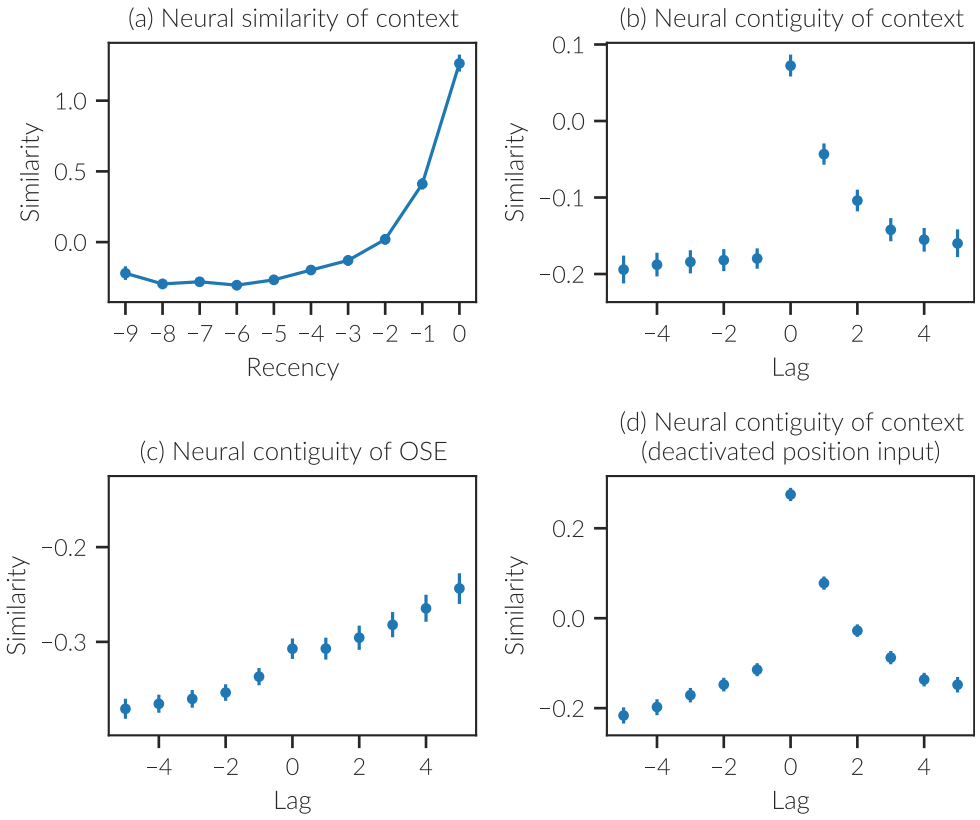


Figure 16. Analyses of model spike trains showing temporal context effects. Compare to figure 6 in Folkerts, Rutishauser, and Howard (2018). All error bars show 95 % confidence intervals, but are often covered by the markers. (a) The similiarity of neural representations decreases rapidly with decreasing recency. (b–d) Each point compares the similarity of the z-scored population vectors of a neuron population during presentation and recognition phase for a given lag. (b) Neurons from the context population. (c) Neurons from the OSE population. (d) Neurons from the context population with deactivated position input during the simulation.

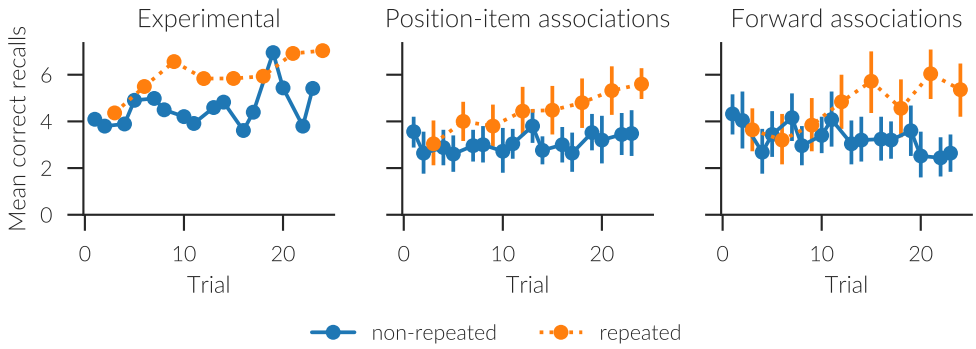


Figure 17. Experimental and model data showing the Hebb repetition effect. Nine item lists were presented and one list was repeated on every third trial. From left to right: experimental data (Hebb, 1961), model data with direct learning of position to item associations, and model data with learning of forward associations. The error bars show 95 % confidence intervals (no confidence intervals were provided for the experimental data).

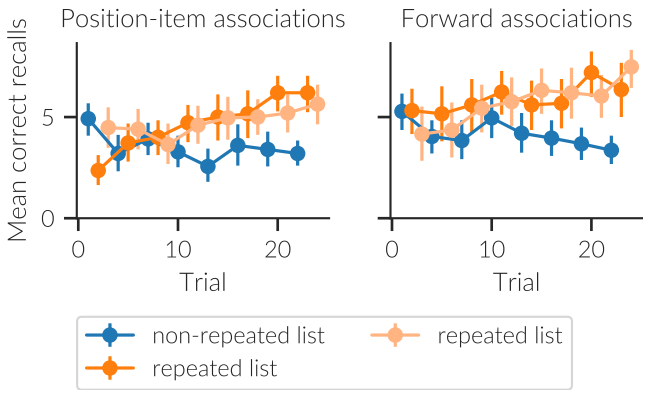


Figure 18. Three lists are shown to the model, two of which are repeated and one of which is not. Both repeated lists show improved performance both in learning of position to item associations, and learning of forward associations. The error bars show 95 % confidence intervals.

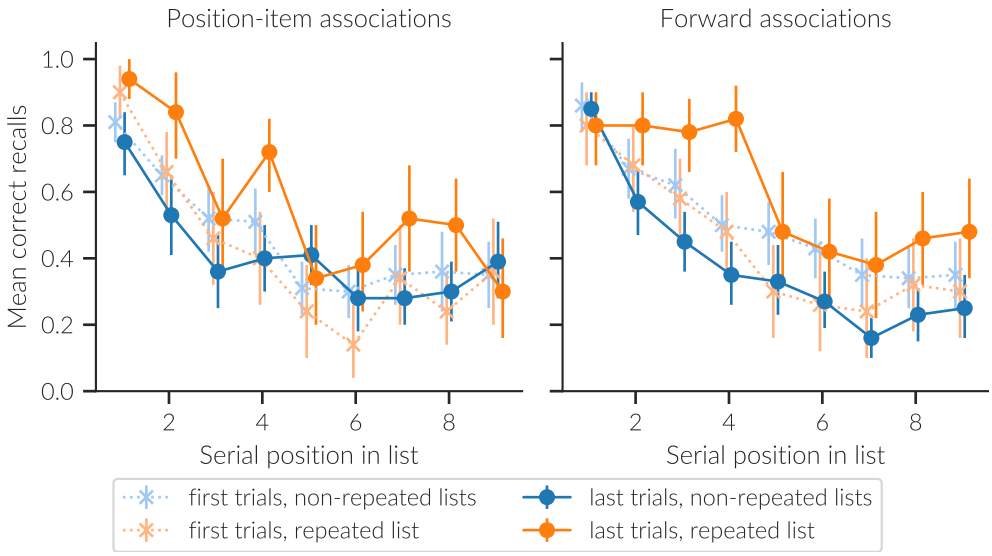


Figure 19. Model data showing the effects of repeating the first four items in a list. First trials are averaged over the first six trials, and last trials are averaged over the last six trials. Partially repeated lists show no effect in position-item associations, although there is an improvement in forward associations. The error bars show 95 % confidence intervals (no confidence intervals were provided for the experimental data).