An oscillatory neural network model of sequential short-term memory with cyclic connections

Arya Teymourlouei
Department of Computer Science
University of Maryland
College Park, USA
ateymo2@umd.edu

Rodolphe Gentili*
Department of Kinesiology
University of Maryland
College Park, USA
rodolphe@umd.edu

James Purtilo*
Department of Computer Science
University of Maryland
College Park, USA
purtilo@umd.edu

James Reggia*
Department of Computer Science
University of Maryland
College Park, USA
reggia@umd.edu

***Abstract*—One of the important neural processes which leads to executive functions in the human brain is the storage of a small amount of information, known as either short-term memory (STM) or working memory. If the memories have a serial order, this extends to sequential STM. Constructing models of sequential STM is an important problem, but is challenging due to the brain's complexity. Prior work used oscillatory neural networks to model the encoding and recall of approximately very few (1 or 2) memories. However, it is known that human STM capacity can range from 3-9 items. Existing neural network models of STM do not model this behavior. In the current research, we show that modifying a prior model by including cyclic connections during the learning process leads to surprising implications. Our model performed significantly better than in the past and was able to closely match human behavioral data from a very recent study. Cyclic connections also produced better recency effects and could model how the mode of stimulus presentation impacted sequence recall. This research is consistent with and extends the findings of prior work. The model introduced here may support neural engineering solutions and improve our understanding of human cognitive-motor processes.**

Keywords*— *Oscillatory recurrent neural network, Short-term memory, sequential memory model, asymmetric Hebbian learning, working memory

I. Introduction

The sophisticated capabilities of the human brain arise from the emergent properties which arise from it, notably, high-level cognitive abilities. These are collectively known as the brain's executive function, a core component of which is the mechanism to keep and manipulate recently acquired information, known as working memory (WM) [1, 2]. Some examples of cognitive processes that are built upon WM include decision-making [3], motor execution [4], and cognitive task completion [5]. When the memories have a serial ordering, this is known as sequential WM. Since the manipulation of memory items is highly context-dependent, in this work we focus on only the storage aspect of WM, and specifically refer to this as *short-term memory* (STM). Constructing models of sequential STM is an important problem which can improve our understanding of brain function. The goal of this current research is to build upon existing models of sequential STM to more closely match the results from a very recent experimental study [7].

Prior work has used neural networks to model sequential STM [6]. Such models usually consist of two phases: Learning and recall. During learning, the network is trained by encoding the memories into weighted matrices. This is usually done with Hebbian learning, in which the connection between two nodes is strengthened when they are simultaneously active [8]. In the recall phase, the network is initialized to a random state and run for some number of timesteps. When the neural network has oscillatory dynamics, its state can transition between the stored memories during the simulation. In particular, when the weights matrix is temporally asymmetric, the network can transition from one stored memory to the next in the order in which the memories were learned. This makes it possible to model sequential STM. Prior work used this model to successfully model human behavioral data from the *n*-back task [6].

However, the performance of this model has limitations. It was shown that when the model was trained with sequences of six items, it was only able to recall 2.26 items on average in the best case [6]. This also means that the neural network could not model instances where the entire sequence was recalled. For the prior work, this was not an issue because the human behavioral data being modeled came from a very difficult task, so the performance of the participants was similar to that of the model. However, for more typical memory recall, prior work estimated that memory capacity is approximately 5-9 items [9]. This was later refined to 3-5 items [10]. The existing oscillatory neural network cannot model this kind of behavior.

To address the limitations of the oscillatory neural network model of sequential STM, a simple modification to the existing learning rule is made. The existing model learned two weights matrices, W and V. The matrix W is the usual Hebbian weights computed by taking the outer product of each memory, and it is symmetric. V is a temporally asymmetric weights matrix which captures the transitions between the stimuli in the sequence. For example, if the sequence $\{1, 9, 3, 5, 2\}$ had to be learned, V would encode the transitions $1 \rightarrow 9, 9 \rightarrow 3, 3 \rightarrow 5$, and $5 \rightarrow 2$. In our model, we add one more step to the construction of V. In this step, a *cyclic connection* is formed between the final memory and the first one. In the example above, this corresponds to the connection $2 \rightarrow 1$. We hypothesize that this will increase the number of memories the model can recall. Why? Consider the case where the network converges to the final learned memory during a simulation. In the existing model, it has nowhere else to go. That is, V does not encode any transition from the final memory. But, in our model, there is a cyclic connection to the first memory. So, the network is surprisingly much more likely to not get "stuck" in the final memory. A numerical example of this is illustrated in Fig. 1.

*Co-senior authors.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

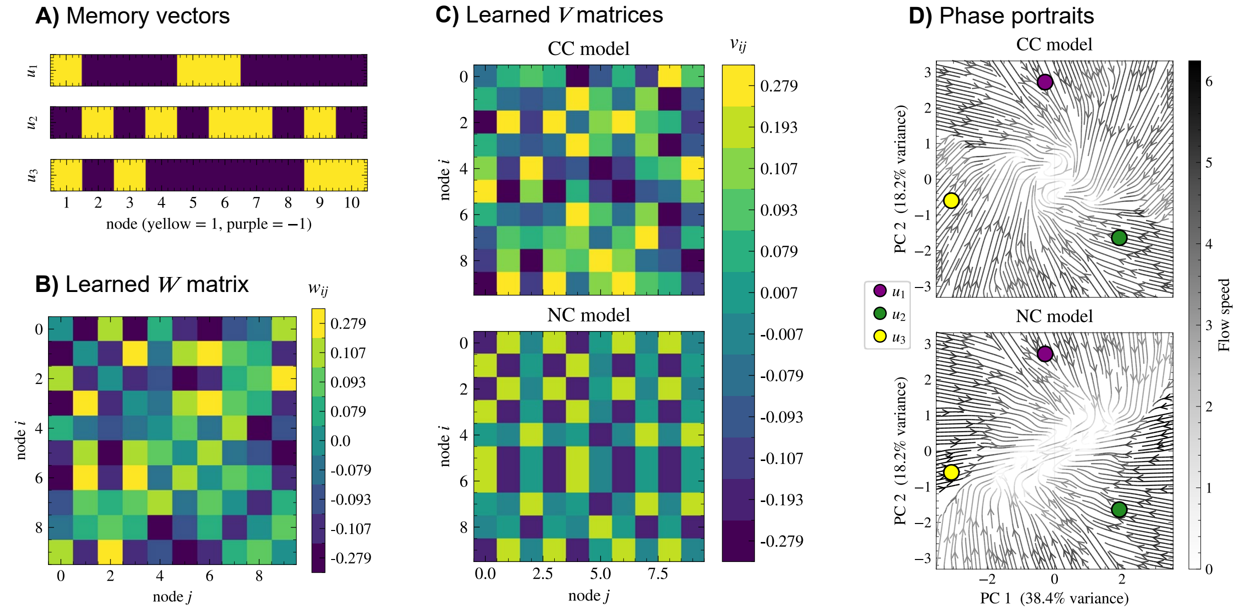

Fig. 1. A toy numerical example illustrating the neural model of sequential short-term memory. The number of nodes is 10, the number of memories is 3 (each denoted by u_i for $i = 1,2,3$), W is the symmetric weights matrix, and V is the temporally asymmetric weights matrix. **A)** The memory vectors. **B)** W after learning. **C)** V after learning, with a cyclic connection (CC) added or not (NC). The cyclic connection encodes a transition from u_3 to u_1. **D)** Phase portraits which show the network's trajectory from 6000 initial random states after one step, with direction indicated by the arrows' orientation and speed indicated by the arrows' color. States are mapped to 2 dimensions (2-d) using principal component (PC) analysis. Legend denotes locations of stored memories in 2-d space.

Although the inclusion of cyclic connections could be seen as a modest change, and algorithmically it is, we will show that it has striking consequences. To realize the significance of the modification, consider the following observation. During the recall phase, when the network is simulated, the activation state is initially random. However, this does not imply that there is an equally likely chance that the network will reach any of the stored memories first. Rather, the network is much more likely to converge to the memory which was learned last (i.e., the one with the largest ordinal position), as demonstrated in prior work [6]. Given this observation, it becomes clear that the existing STM model will not be able to recall the entire sequence, because the network's state rarely converges to the first learned pattern (which is where a serial recall would start). Adding a transition from the final memory back to the first will give the network a much better chance to converge to the latter. From there, it can drive through the stored patterns in the serial order it was trained. Computational evidence for this intuition is provided in the results by studying recall accuracy as a function of the ordinal position of the learned memory.

Cycles/loops are known to exist in the brain's neural circuits. Some examples are cortico-thalamic loops, cortico-cerebellar loops, and cortico-basal ganglia loops. Theories have been proposed to explain how loop structures in the brain account for executive function and other cognitive-motor processes. For example, a recent article proposed that rodent brain function is organized into a hierarchy of loops, where WM is mediated through the 2nd-highest loop which contains the dorsomedial prefrontal cortex and the dorsomedial striatum [11]. Another article studied spiking neural network models of sequential memory, controlling the length of the simulated brain loop structure and the delay in information transmission [12]. From this perspective, there is a biological justification of the cyclic connections proposed in our neural model, and this approach follows from some of the existing theory and experimental findings regarding STM.

In this research, we evaluate the computational and statistical properties of the oscillatory neural network model of sequential STM which incorporates cyclic connections. This model is compared to the existing one in [6]. The methodology formally describes both models and explains the procedure for evaluating recall. In the results, the model parameters are studied with respect to the performance of the model. Finally, it is shown that by modulating a single parameter of the new model, it can be fit to human behavioral data from a recent investigation.

II. Methodology

A. Notation

Matrices are denoted by uppercase letters. Vectors are denoted by lowercase letters with overhead arrows (e.g., $\vec{x}$). Scalar values are denoted by lowercase letters without any overhead arrows. Also, for an arbitrary matrix M, m_{ij} refers to the entry in M at row i, column j. For a given vector $\vec{x}$, x_i refers to the ith entry in $\vec{x}$.

B. Neural Network Model

The model used here follows prior work [6]. We use a fully connected oscillatory neural network of n nodes, where n is set to 180 in all simulations. The term oscillatory refers to the fact that the network's output dynamics converge to a limit cycle which consists of the stored memories. Note that spurious states are possible in these cycles. The dynamics of the network are described by the vector $\vec{a}^t = (a_1, a_2, \ldots, a_n)$, where $a_i^t \in \{-1,1\}$ is the value of node i at time t.

When considering a sequential STM process in the brain, there are two primary steps. In the first step, the memories are learned. In the second step, the memories are recalled in the correct serial order. As the length of the sequence increases, it becomes more challenging to recall the sequence correctly.

To model this process, the network is operated in a two-phase manner which mirrors the process described before. The model is illustrated in Fig. 2. In the first step, the network learns m patterns. The value of $m = 7$ is used in all simulations so that the dynamics of the model can be compared to human behavioral data acquired in a very recent study [7], where participants had to memorize sequences of 7 digits.

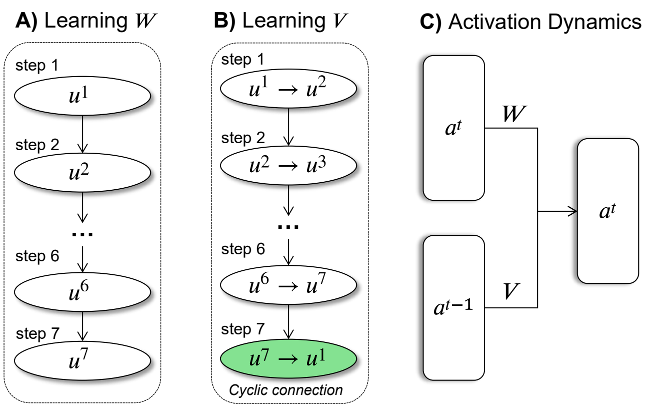

Fig. 2. The neural model used in this work. A) The matrix W is constructed via Hebbian learning with weight decay, and memories are learned in sequential order. B) The matrix V is learned via temporally asymmetric Hebbian learning, with the use of a cyclic connection as the final step. C) After learning, the state of the network a^t is computed at each timestep t using W and V.

Denote the memories to be learned as $\vec{u}^p$, for $p = 1,2, \dots, m$. These are encoded into the $n \times n$ matrices W and V. Both matrices are constructed using iterative algorithms which run for m steps, learning one pattern each step. Both algorithms start by initializing all entries of the matrices equal to zero. The matrix W is analogous to the weights used in typical neural models of associative memory (e.g., Hopfield network). Define W^p to be the matrix W at iteration p of the algorithm. Then, for each step $p = 1,2, \dots, m$, we update W^p with the following rule:

$$W^{p+1} = (1 - k_d)W^p + \mu \cdot \vec{u}^p(\vec{u}^p)^T \tag{1}$$

where k_d is a decay rate (such that $0 \le k_d < 1$), $\mu = 1/n$ is the learning rate and $(\cdot)^T$ denotes the vector transpose. This is effectively Hebbian learning with possible decay. Note that the relative effects of decay versus interference in forgetting have long been a fundamental issue in psychology [13]. At the end of the algorithm, all of the diagonal entries of W are set to zero, so there are no self-connections.

The matrix V, which has the temporally asymmetric weights that are responsible for the ordering of the stimuli, is constructed with a similar rule. As before, let V^p be the iteration of V at step p. Then, for $p = 2,3, \dots, m$,

$$V^{p+1} = (1 - k_d)V^p + \mu \cdot \vec{u}^p(\vec{u}^{p-1})^T \tag{2}$$

with symbols analogous to (1). Note that we do not start with $p = 1$ because $\vec{u}^{-1}$ is undefined.

It is also possible to incorporate *cyclic connections* to V. A priori, one might expect this to have little effect, but we show in the following that it makes a huge difference in the behavior of the model, and it has not been explored significantly in previous work [6, 14]. This is accomplished with one additional step to the algorithm above:

$$V^{m+1} = (1 - k_d)V^m + \mu \cdot \vec{u}^0(\vec{u}^m)^T \tag{3}$$

With this extra step, a transition from the final memory to the first memory is created. This means that during a simulation, if the network recalls the final memory at some timestep t, it is more likely to transition back to the first memory at $t + 1$. Note also that the cyclic approach (3) to the construction of V involves m steps, while the non-cyclic approach involves $m - 1$ steps.

After W and V are constructed, the first phase of the model's operation is finished. In the second phase, the neural network is simulated for t_m timesteps ($t_m = 200$ in all experiments). The initial state of the network, $\vec{a}^0$, is set randomly. Then, for each step $t = 1,2, \dots, t_m$, the value of each node is asynchronously updated in a random order using the following rule:

$$a_i^t = \mathrm{sgn}\left(\sum_{j=1}^{n}\left(\beta_1 w_{ij} a_j^t + \beta_2 v_{ij} a_j^{t-1}\right) - \theta_i^t\right) \tag{4}$$

where $\mathrm{sgn}(\cdot)$ is the sign function, β_1 is the strength of influence of the contribution from W, β_2 is the strength of influence of the contribution from V, and θ_i^t is a dynamic threshold. The use of the threshold allows the network to oscillate between different states, ideally corresponding to stored memories as illustrated below. Since V is an asymmetric matrix, when β_2 is sufficiently large, it is probable that the network will not oscillate randomly between states but rather be driven from one stored memory to the next, in the serial order that they were presented. This allows the neural network to serve as a model sequential STM. The threshold decays with time, and is defined as follows:

$$\theta_i^{t+1} = \begin{cases} (1 - k_\theta)\theta_i^t & \text{if } a_i^{t-1} \neq a_i^t \\ (1 - k_\theta)\theta_i^t + k_w a_i^t & \text{otherwise} \end{cases} \tag{5}$$

where k_θ and k_w are constants such that $0 < k_\theta < k_w < 1$. The idea of (5) is that if the activation of node i does not change from one timestep to the next, then the threshold value for the next timestep is increased by $k_w a_i^t$. This pushes the threshold closer to the state of node i, so it is more likely to change state. The values of $k_\theta = 0.09$ and $k_w = 0.175$ are always used in all experiments. These are the same as in prior work [6]. The values of β_1, β_2, and k_d vary by the experiment.

C. Model Recall Assessment

After the second phase is completed, there are a total of t_m network states. For each state of the network, denoted $\vec{a}^t$ (at step t), the Hamming distance between $\vec{a}^t$ and each of the stored memories is computed. If this distance is zero for some memory $\vec{u}^p$ ($p = 1, \dots, m$), i.e., $\vec{a}^t = \vec{u}^p$, we say that $\vec{u}^p$ was *recalled* at step t.

A list $\vec{\ell}$ of length t_m is created such that

$$\ell_i = \begin{cases} p \text{ if memory } p \text{ recalled at step } i \\ -1 \text{ otherwise} \end{cases} \tag{6}$$

for $i = 1, \ldots, t_m$. Then, all the possible length-m sub-lists of $\vec{\ell}$ are examined. If there is at least one sub-list where the entire sequence is recalled, i.e., m consecutive timesteps such that the memories are recalled in the correct serial order in those steps, then we say that the model recalled the sequence correctly. For example, if $m = 4$, then the sub-list $[1,2,3,4]$ would indicate a perfect recall. If this does not occur, the "best" sub-list is identified, which simply is the sub-list which has the most memories out of m recalled in the correct serial order. As an example, if the best sub-list was $[1, -1,3,4]$, 3 of 4 memories were correctly recalled, but the entire sequence was not recalled.

D. Empirical Behavorial Dataset

A recent study examined how sequential WM is impacted by the mode in which the sequence is presented to the participant [7]. The task involved learning a sequence of 7 digits and recalling them in the correct serial order. Four modes of presentation were considered. The first was simultaneous, where the entire sequence was shown once (for 2800 ms). For the remaining three, the digits were presented sequentially either fast (each digit was shown for 400 ms), slow (each digit was shown for 1000 ms per digit), and fast with delay (each digit was shown for 400 ms followed by a 600 ms delay period in-between). The study consisted of 35 participants (5 removed due to data issues). Each participant completed 50 trials for each of the four modes of sequence presentation. In the results, the recall performance of the participants is compared to that of the model.

III. Results

All figures were made with the Python plotting packages Matplotlib [15] and/or Seaborn [16].

A. Experiment One: Beta Parameters

In the first experiment, the ability of the models to recall the learned memories is assessed when modulating β_1, β_2, and k_d. To this end, a grid search is conducted over the following values of β_1 and β_2: 0.05, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5. For k_d, the following values are tested: 0, 0.04, 0.08. For each combination, the model learns a sequence of 7 random patterns. The network is initialized to a random state and the simulation is run. At the end the number of correctly recalled memories (out of 7) is computed. This process is repeated 100 times and the average score is recorded. The scores are shown as a contour plot in Fig. 3, for two versions of the model. In the first version, V has cyclic connections (henceforth known as the CC model). In the second, V does not (the NC model).

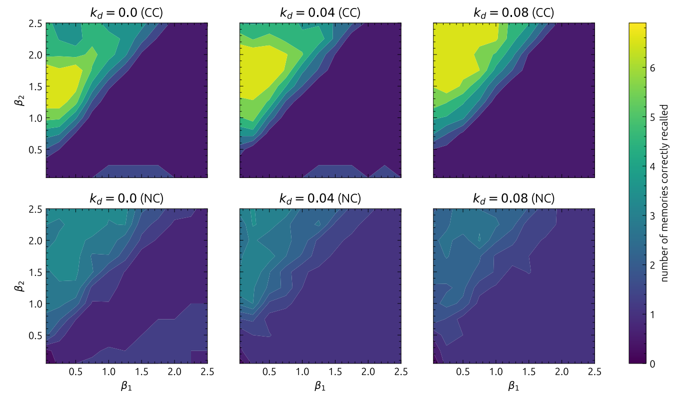

Fig. 3. Average number of memories (out of seven learned) recalled by the model when using cyclic connections (top row, denoted by CC) or not (bottom row, denoted by NC).

The results of Fig. 3 demonstrate that the inclusion of cyclic connections in V significantly improves the performance of the model. In fact, the CC model is able to recall all seven memories on average in the best case (the yellow colored regions in Fig. 3). This is achieved for approximately $\beta_1 = 0.05$ and $\beta_2 = 1.5$, for any k_d. For the NC model, the model can only recall 2-3 memories at best, with this performance being achieved for similar choices of β_1 and β_2. It is also observed that for the CC model, as k_d increases, the model performs better for larger β_2. These results suggest that in general, for the CC model, the recall performance of the model improves when β_2 is much larger than β_1. This is consistent with prior NC work, which found that the largest average number of items recalled from a 6-input sequence was achieved for $\beta_2 > \beta_1$ [6].

B. Experiment Two: Weight Decay

In the second experiment, the impact of the weight decay k_d is thoroughly investigated with respect to the recall ability of the model. This time, based on the findings of the first experiment, the model always uses $\beta_1 = 0.05$ and $\beta_2 = 1.5$. The same process for learning and simulating the network is done as in the first experiment, except the average score is computed across 1000 repetitions this time. The average number of correctly recalled memories (from the 7 learned) is shown in Fig. 4. This is the same quantity which was computed in Fig. 3. Note that in Fig. 4, the error bars for some points are too small to see.

The results in Fig. 4 show that the CC model has much better performance than the NC model. This is consistent with the finding from the first experiment. In the best case ($k_d = 0.02$), the CC model recalls 6.8 memories on average, which is very close to the 7 learned. However, the NC model does not recall more than 3 memories on average. For both variants of the model, performance decreases as the weight decay increases. These results suggest that smaller values of k_d leads to better performance, which generally agrees with prior work, although there are some differences due to variation in the choice of β_1 and β_2 [6]. Further, Fig. 3 shows that high performance can still be attained for larger k_d if β_2 is scaled proportionally.

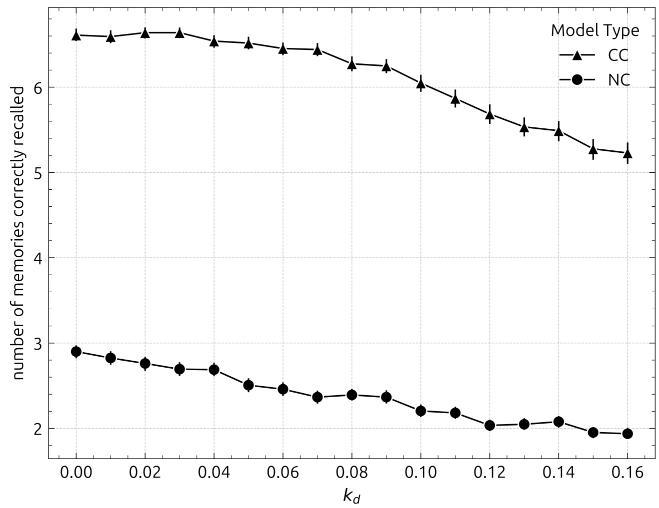

Fig. 4. The average number of memories recalled by the model for a single sequence (out of 7 learned) when using cyclic connections (CC) or not (NC) as a function of the weight decay parameter k_d. The error bars represent a 95% confidence interval.

Next, the average fraction of correctly recalled sequences is studied as a function of k_d. This quantity measures how often the model recalled all 7 memories. These results are taken from the exact same experiment as in Fig. 4, the quantity being measured is just different. The results are shown in Fig. 5. The error bars are not visible in Fig. 5 for the NC model since it was never able to correctly recall a sequence.

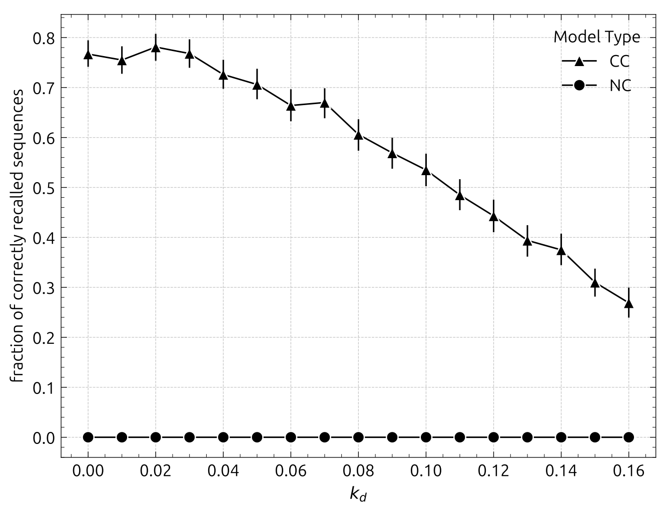

Fig. 5. The average fraction of correctly recalled sequences by the model when using cyclic connections (CC) or not (NC) as a function of the weight decay parameter k_d. The error bars represent a 95% confidence interval.

In Fig. 5, it is seen that for the CC model, in the best case ($k_d = 0.02$), the model recalls the entire sequence correctly nearly 80% of the time (~800 of the 1000 repetitions). In stark contrast, the NC model is not able to correctly recall the entire sequence even once. As before, performance for the CC model decreases with increasing k_d.

In Fig. 6, the results of the second experiment are further investigated by studying the recall accuracy with respect to the individual memories which were learned. It is demonstrated that for the CC model, when $k_d \leq 0.04$, all memories are recalled with over 80% accuracy. When k_d is increased, the accuracy of recall decreases for all the memories in the sequence except for the final memory. It is observed that the recall accuracy becomes disproportionally worse for the earlier memories compared to the ones later in the sequence as k_d increases. Based on these observations, it is possible that the neural network with cyclic connections for larger k_d can be seen as modeling the recency effect during memory recall much more than the NC model. From a computational perspective, the behavior of the CC model in Fig. 6 could possibly be because the effect of weight decay is felt more strongly on the memories with an earlier ordinal position (except for the first memory, due to the cyclic connection, which is not impacted by decay).

Fig. 6 also shows that for the NC model, the recall accuracy increases nonlinearly with the ordinal position of the memory in the learned sequence. The accuracy of recall worsens as k_d increases. These are consistent with the findings of prior work [6, 14]. In a sense, this neural network can also be seen as a model of the recency effect, although differently than before.

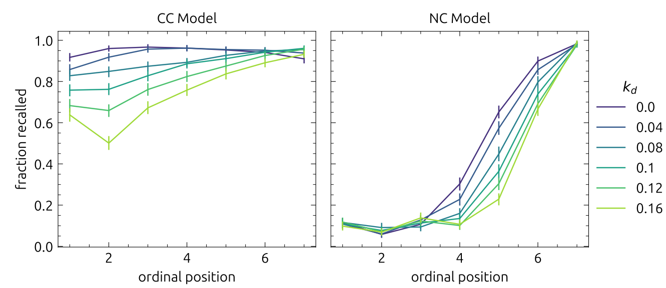

Fig. 6. The average fraction of correctly recalled memories by the model when using cyclic connections (CC) or not (NC) as a function of the ordinal position of the memory. Different values of k_d are considered and indicated in the legend. The error bars represent a 95% confidence interval.

C. *Experiment Three: Empirical Data Analysis*

In the third experiment, the model is fit to empirical behavioral data collected from a human WM study. The data for the model is the same as used from the second experiment. The fit is done by simply visually comparing the results of Fig. 4 and Fig. 5 to the empirical data and choosing k_d which minimizes the error. The results are shown in Fig. 7.

The model was fit to the data by decreasing k_d to improve the recall performance, and increasing k_d to worsen the model's performance. In Fig. 7, for the simultaneous condition, $k_d = 0.05$ was used. The slow condition used $k_d = 0.09$ and the fast condition with delay used $k_d = 0.10$. For the fast condition, $k_d = 0.15$ was used. Therefore, it can be concluded that simple modulation of the k_d parameter can allow the model to fit remarkably well to the empirical observations. This is consistent with the findings of prior work [6, 14].

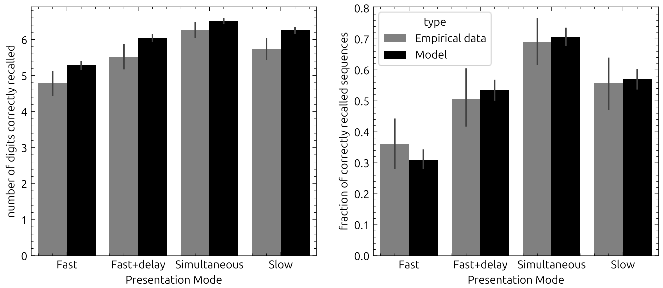

Fig. 7. Empirical data from a human performance study and corresponding fits from the CC model. *Left plot*: The number of digits correctly recalled in a single sequence (of length 7). *Right plot*: The fraction of correctly recalled sequences. Error bars denote 95% confidence interval.

In addition, the fraction of correctly recalled sequences as a function of the ordinal position is presented in Fig. 8. It is shown that on average, the mode of simultaneous presentation affects the best recall accuracy. The worst performance is observed when the sequence is presented digit-by-digit in a fast manner. When the curves in Fig. 8 are compared to the ones in Fig. 6, there are several observations that can be noted. First, the results from the CC model for $0.10 \leq k_d \leq 0.16$ are similar to the results in Fig. 8 in the sense that both are skewed U-shaped curves which are typically associated with the serial position effect. However, the curves from the model are skewed such that memories with a larger ordinal position have better recall accuracy, whereas the inverse is true for the results from the human behavioral data. This implies that the model assumed a stronger recency effect, whereas the empirical data had a stronger primacy effect. Some implications of this difference are explained in the discussion.

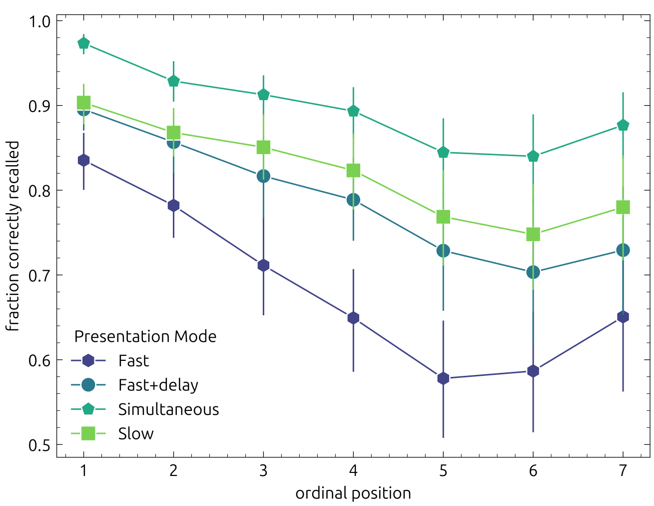

Fig. 8. The average fraction of correctly recalled digits as a function of the ordinal position of the digit from human behavorial data. The legend indicates the presentation mode. Error bars represent a 95% confidence interval.

IV. Discussion

Sequential STM is one of the foundations for high-level cognitive abilities in humans. There is therefore no doubt that the modeling sequential STM as is done here is an important research problem. In this work, an oscillatory neural network with temporally asymmetric Hebbian learning was adjusted to incorporate cyclic connections with the goal of improving the ability of the model to recall sequences which were of longer length than what was considered in prior work [6, 14]. The results are surprising in that they showed that the simple inclusion of cyclic connections significantly improved the recall accuracy of individual memories within the sequence, as well as facilitating the recall of entire sequences of length seven. The latter was not possible at all without cyclic connections.

This neural network used in this work is closely related to the Hopfield network [17]. Recently, a variant of the traditional Hopfield network, known as the Modern Hopfield Network (MHN), was introduced. The MHN provides significantly larger memory capacity than the traditional version [18], based mainly around reducing the amount of interference or crosstalk between memories. A very recent work related to this paper introduced a model of sequential memory which used MHNs as a basis, known as the DenseNet [19]. The DenseNet is different from our model in that the former uses a nonlinear function in the activation dynamics to reduce the overlap between the state of the network and learned memories (ours does not do this). It was shown that the DenseNet could perfectly recall sequences which have a length m that is polynomial (or even exponential) relative to the size of the network n. This model undoubtedly has a larger sequence capacity than the CC model studied in this work. However, a significant limitation of the DenseNet is that it is biologically unrealistic, as it implies that nodes must have a huge (infinite, if exponential nonlinearity is used) number of connections with other nodes.

The main limitation of this work was that it was not possible to model the primacy effect which was observed in the human behavioral data. Instead, the neural network was only able to model the recency effect (for some choices of k_d). In this sense, the neural network is not yet fully a robust model of the serial position effect. However, it should be noted that this was not the goal of the research. Although it was possible to modulate the recency effect through k_d, this also had limitations in its robustness. The neural network used in this work was limited in that it used a fully connected network (the brain largely relies on local and modular connectivity) and that it was only equipped to handle discrete, binary-valued memories. Finally, this work only considered behavioral data from a digit recall task. This to keep consistent with the prior work in this area [6, 14], but it should be noted that WM can be dependent upon the type of stimuli (e.g., letters, words, images). Recent work has underscored the impact of complexity of memory items on WM capacity [20].

Future work should further investigate the computational properties of the CC model to more completely assess the impact of cyclic connections. Modifications to the model can also help address the limitations discussed here, particularly, the creation of an effective mechanism that more robustly models the serial position effect and controls the associated dynamics of the primacy and recency effect. Also, integrating more biologically plausible components to the model (e.g., support for continuous-valued memories) can help to improve the realism of the model. Finally, it would be interesting to assess to what extent the neural network can be used to model chunking dynamics in STM as well as human memory dynamics in more rich settings, such as group meetings and collaborative activities.

ACKNOWLEDGMENT

For aiding the development of software used in this research, we express our thanks to Ryan DeCamp, Abdullatif Elmuaqqat, Shaun Fernandes, Pranav Khatri, and Vihaan Motwani. We also thank Jayesh Jayashankar for his feedback on the research. The code used for Figure 1 was initially created by Claude Sonnet 4.6. It was reviewed and then modified for the paper's purposes.

REFERENCES

[1] D'Esposito, M., & Grossman, M. (1996). The physiological basis of executive function and working memory. *The Neuroscientist*, *2*(6), 345-352.

[2] Mesulam, M. M. (1998). From sensation to cognition. *Brain: A Journal of Neurology*, *121*(6), 1013-1052.

[3] Del Missier, F., Mäntylä, T., Hansson, P., Bruine de Bruin, W., Parker, A. M., & Nilsson, L. G. (2013). The multifold relationship between memory and decision making: an individual-differences study. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *39*(5), 1344.

[4] Courtine, G., Papaxanthis, C., Gentili, R., & Pozzo, T. (2004). Gait-dependent motor memory facilitation in covert movement execution. *Cognitive Brain Research*, *22*(1), 67-75.

[5] Sohn, Y. W., & Doane, S. M. (2003). Roles of working memory capacity and long-term working memory skill in complex task performance. *Memory & Cognition*, *31*(3), 458-466.

[6] Sylvester, J. C., Reggia, J. A., Weems, S. A., & Bunting, M. (2010, August). A temporally asymmetric hebbian network for sequential working memory. In *10th Int'l Conf. Cognitive Modeling* (pp. 241-246).

[7] Kosachenko, A. I., Syttykov, D. I., Tarasov, D. A., Kotyusov, A. I., Kasanov, D., Kotchoubey, B., & Pavlov, Y. G. (2025). Alpha activity during verbal working memory retention depends on the stimulus presentation mode. *bioRxiv*, 2025-05.

[8] Hebb, D. O. (2005). The organization of behavior: A neuropsychological theory. *Psychology Press*.

[9] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81.

[10] Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87-114.

[11] Rusu, S. I., & Pennartz, C. M. (2020). Learning, memory and consolidation mechanisms for behavioral control in hierarchically organized cortico‐basal ganglia systems. *Hippocampus*, *30*(1), 73-98.

[12] Sihn, D., & Kim, S. P. (2024). A neural basis for learning sequential memory in brain loop structures. *Frontiers in Computational Neuroscience*, *18*, 1421458.

[13] Altmann, E. M., & Gray, W. D. (2002). Forgetting to remember: The functional relationship of decay and interference. *Psychological Science*, *13*(1), 27-33.

[14] Reggia, J. A., Sylvester, J., Weems, S. A., & Bunting, M. F. (2009). A Simple Oscillatory Short-Term Memory. In *AAAI Fall Symposium: Biologically Inspired Cognitive Architectures.*

[15] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(03), 90-95.

[16] Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021.

[17] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554-2558.

[18] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., ... & Hochreiter, S. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

[19] Chaudhry, H. T., Zavatone-Veth, J. A., Krotov, D., & Pehlevan, C. (2024). Long sequence hopfield memory. *Journal of Statistical Mechanics: Theory and Experiment*, *2024*(10), 104024.

[20] Lin, W., Lv, C., Liao, J., Hu, Y., Liu, Y., & Lin, J. (2024). Feature versus object in interpreting working memory capacity. *Npj Science of Learning*, *9*(1), 67.