A computational model for simulating the future using a memory timeline

Zoran Tiganj (ztiganj@iu.edu)
Department of Computer Science & Department of Psychological and Brain Sciences, Indiana University

Wei Tang (wt1@iu.edu)
Department of Computer Science & Department of Psychological and Brain Sciences, Indiana University

Marc W. Howard (marc777@bu.edu)
Department of Psychological and Brain Sciences, Boston University

Abstract

The ability to learn temporal relationships and use that knowledge to simulate future events is among the most remarkable aspects of cognition. Recently introduced behavioral task called Judgment of Imminence (JOI) combined with a well-known Judgment of Recency (JOR) task pointed to a remarkable symmetry between the temporal organization of memory and prediction. The data were consistent with the hypothesis that both memory and prediction can be organized as a compressed mental timeline. This means that the past and future can be remembered or simulated sequentially relative to the present. The compression implies that events closer to the present, regardless of whether they are in the past or in the future, were represented more accurately than those further from the present. Here we used the existing JOR model based on a compressed memory timeline to build an associative representation that can learn the temporal relationships and create a timeline of the future, which mirrors the timeline of the past. We show that this approach can simultaneously account for response times and accuracy in both JOR and JOI. This work provides a time-local neural-level mechanistic account for how the temporal organization of the memory can be used to learn the temporal structure of the world and simulate the future in an efficient manner as a compressed mental timeline.

Keywords: Associative memory, Timeline, Scale-invariance, Prediction, Sequence learning.

Introduction

The brain's remarkable ability to learn temporal relationships and use them to predict the future has been in the focus of many cognitive scientists. Here we evaluate a computational hypothesis that proposes using associative memory to learn temporal relationships and construct an estimate of the future (Howard, Shankar, Aue, & Criss, 2015; Shankar, Singh, & Howard, 2016; Tiganj, Gershman, Sederberg, & Howard, 2019). In a nutshell, this hypothesis makes two major assumptions: 1) Memory of the recent past is maintained as a compressed neural timeline: the memory about what happened when is carried in a population of sequentially activated, stimulus-specific neurons. Importantly, the more recent past is represented with more neurons. Those neurons have narrower firing fields, resulting in a gradual decay of temporal resolution from more recent to more distant past. This memory representation was used to account for different behavioral experiments, including time estimation and recency judgments (Howard et al., 2015; Tiganj, Cruzado, & Howard, 2019). Sequentially activated neurons, called time cells, were observed in different parts of the brain, especially in the hippocampus (MacDonald, Lepage, Eden, & Eichenbaum, 2011; Pastalkova, Itskov, Amarasingham, & Buzsaki, 2008; Tiganj, Kim, Jung, & Howard, 2015). Neural recordings also suggest that time cells are stimulus-specific (different stimuli activate different time cells) (Tiganj, Cromer, Roy, Miller, & Howard, 2017; Cruzado, Tiganj, Brincat, Miller, & Howard, 2020), supporting the existence of a memory timeline. 2) Input stimuli are associated with the memory timeline through Hebbian learning. The associations store the average temporal history for every stimulus. The temporal history of a stimulus contains information about which stimuli preceded that stimulus and when. The average history is then used to compute the average future (Tiganj, Gershman, et al., 2019).

To evaluate the proposed hypothesis here we focus on the data published in a recent behavioral study that examined the similarity between memory and prediction (Tiganj, Singh, Esfahani, & Howard, 2020). In this study, participants completed two tasks, a well-established relative judgment of recency (JOR) and a new task, called judgment of imminence (JOI). In JOR participants are presented with a sequence of stimuli followed by a probe consisting of two letters from the sequence. The participants have to select the letter that was presented more recently (Figure 1a). The study confirmed the classic finding that in JOR response time (RT) in correct trials depends on the lag to the more recent probe but not on the lag to the less recent probe (Hacker, 1980; Hockley, 1984). This finding is consistent with a self-terminating backward scan along a temporally-organized memory representation (Hacker, 1980; Hockley, 1984; McElree & Dosher, 1993). In addition, Tiganj et al. (2020) found that in correct trials, the RT scales sublinearly with the lag of the more recent probe. This is consistent with the hypothesis that memory is organized as a compressed neural timeline, as demonstrated in a computational model (Howard et al., 2015; Tiganj, Cruzado, & Howard, 2019).

JOI task was designed as a future-time analog of JOR. By analogy to the way the JOR evaluates participants' ability to judge the relative time at which past events occurred, this new paradigm tested participants' ability to judge the imminence of future events over a scale of a few seconds. The relative JOR task requires the participant to select the probe item from the previous list that was presented closer to the present. In contrast, the JOI paradigm asks participants to select the future probe item that *is anticipated* closer to the present (Figure 1b). Here we review the computational model for JOR described in Howard et al. (2015); Tiganj, Cruzado, and Howard

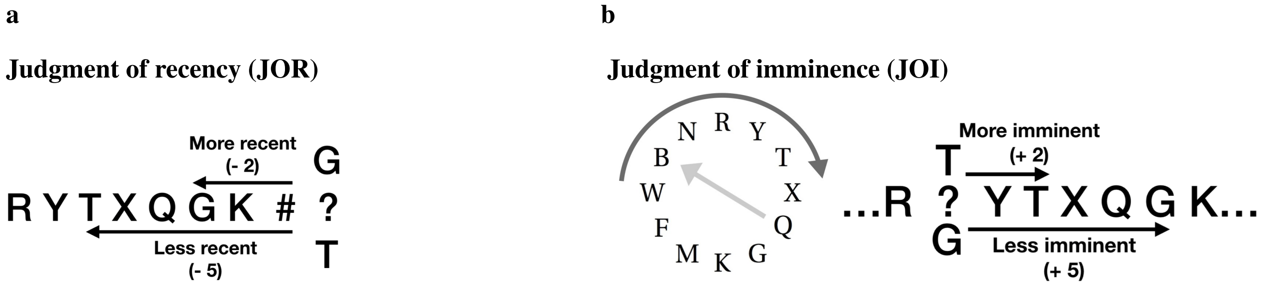

Figure 1: **Judgment of recency (JOR) and judgment of imminence (JOI) behavioral tasks. a.** Schematic of the JOR task. Participants are shown a list of letters (such as RYT. . .) followed by a probe containing two letters from the list (here, G and T). Participants choose the probe item that was experienced more recently. In this example, probe G is the correct answer. Lag of each probe is defined as the number of steps backward in the list necessary to find the probe. **b.** Schematic of the JOI task. In JOI, participants learn a probabilistic sequence. The sequence usually follows a predictable sequence (clockwise movement along the circle) with occasional random jumps (light-grey arrow). After learning these transition probabilities, the sequence is interrupted by a probe containing two letters (here, G and T). Participants choose the letter that is likely to occur sooner. In this example, the letter T is the correct answer. Lag for each probe is defined as the number of steps forward at which one would expect to find the probe if the sequence continued along its most likely path. From (Tiganj et al., 2020).

(2019) and develop a computational model for JOI that drives inspiration from previous theoretical work (Tiganj, Gershman, et al., 2019). We will show that the same computational framework can model the results from both tasks.

Description of the model

We model JOR and JOI using the same computational framework. First, for JOR we construct a logarithmically-compressed memory timeline. Then for JOI we use that timeline to form associations between past and present.

Compressed memory timeline

The approach we used to construct the memory timeline is equivalent to the previous work (Shankar & Howard, 2012; Howard et al., 2015; Tiganj, Cruzado, & Howard, 2019) so here we will only provide a brief review. We encode each stimulus as a one-hot vector $\mathbf{f}$. For instance, for a list of 7 letters, the length of $\mathbf{f}$ is equal to 7. Each element in $\mathbf{f}$ is then fed into a two-layer recurrent network with analytically computed weights. The first layer is recurrent and for the i-th element of the input, vector $\mathbf{f}$ has the following dynamics:

$$\frac{d\mathbf{F}^{(i)}_{\mathbf{s}}(t)}{dt} = -\mathbf{s}\mathbf{F}^{(i)}_{\mathbf{s}}(t) + \mathbf{f}^{(i)}(t). \tag{1}$$

Here $\mathbf{s}$ is an N long vector, where N is the number of neurons in $\mathbf{F}^{(i)}_{\mathbf{s}}$. The impulse response of the neurons in the recurrent layer decays exponentially with rate constants $\mathbf{s}$ (Figure 2, middle row). This layer encodes an approximation of the Laplace transform (see Shankar and Howard (2012) for more details). To obtain a timeline that estimates $\mathbf{f}^{(i)}(t' < t)$ we invert the Laplace transfrom using the Post approximation:

$$\tilde{\mathbf{f}}^{(i)}_{\overset{*}{\tau}}(t) = \frac{(-1)^k}{k!}\mathbf{s}^{k+1}\frac{d^k}{ds^k}\mathbf{F}^{(i)}_{\mathbf{s}}(t), \tag{2}$$

where $\overset{*}{\tau} = k/\mathbf{s}$ is a vector of logarithmically spaced values. This results in a bell-shaped impulse response that activates sequentially across units in $\tilde{\mathbf{f}}^{(i)}_{\overset{*}{\tau}}$ with peaks at $\overset{*}{\tau}$ (Figure 2, bottom row). For a sample trial of JOR task, the activity of the nodes in $\mathbf{f}$ and $\tilde{\mathbf{f}}$ is shown in Figure 3a.

Associative memory

To create the associative memory we use the approach described in (Shankar & Howard, 2012; Tiganj, Cruzado, & Howard, 2019). At each time t, associative memory tensor $\mathbf{M}$ is updated with the outer product of the current input state $\mathbf{f}$ and $\tilde{\mathbf{f}}$. Hence $\mathbf{M}$ is a three-tensor. At each moment, $\mathbf{M}$ is updated with the simple Hebbian learning rule:

$$\mathbf{M}_{\overset{*}{\tau}}(t) = \mathbf{M}_{\overset{*}{\tau}}(t-1) + \mathbf{f}(t)\tilde{\mathbf{f}}_{\overset{*}{\tau}}(t), \tag{3}$$

where $\mathbf{f}$ is a column vector and $\tilde{\mathbf{f}}_{\overset{*}{\tau}}$ is a row vector, both of length N and $\mathbf{M}_{\overset{*}{\tau}}$ is N by N matrix. Formation of the associative memory is illustrated in Figure 3b, where the thickness of the green arrows represents the strength of the associations stored in $\mathbf{M}$.

Compressed timeline of the future

The associative memory is used to construct prediction of the future. To achieve this, we followed the approach described in Tiganj, Cruzado, and Howard (2019). $\mathbf{M}$ stores the pairwise temporal relationships between all stimuli subject to logarithmic compression. For instance, if we had two stimuli A and B presented with spacing Δ, $\mathbf{M}$ will store that temporal relationship. Note that the knowledge that A was presented Δ time before B implies that B was presented Δ time after A. In other words, $\mathbf{M}$ stores the average history, but we can use that information to compute the average future. Specifically, multiplying $\mathbf{M}$ from the left with $\mathbf{f}^{(i)}$ will extract the average history of the i-th stimulus (average $\tilde{\mathbf{f}}^{(i)}$). On the other hand multiplying $\mathbf{M}$ from the right with $\mathbf{f}^{(i)}$ will extract the average future of the i-th stimulus, which we label as $\mathbf{p}^{(i)}$:

$$\mathbf{p}_{\overset{*}{\tau}} = \mathbf{M}_{\overset{*}{\tau}}\mathbf{f} \tag{4}$$

Analogous to $\tilde{\mathbf{f}}$, $\mathbf{p}$ is as a 2-D matrix indexed by stimulus identity and $\overset{*}{\tau}$. Note also that instead of learning the average history we could have directly learned the average future via the successor representation (Momennejad & Howard, 2018).

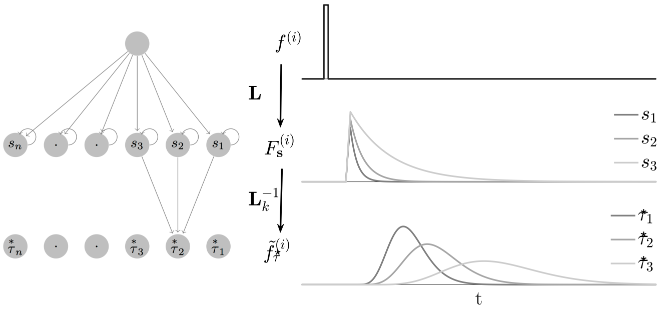

Figure 2: **Schematic of the memory model. Left panel:** The input stimulus f^i feeds into a layer of leaky integrators $\mathbf{F}_{\mathbf{s}}^{(i)}$ that implement a discrete approximation of the Laplace transform. Each neuron in the first layer has a characteristic rate constant s_i. $\mathbf{F}_{\mathbf{s}}^{(i)}$ projects onto $\tilde{\mathbf{f}}_{\overset{*}{\tau}}^{(i)}$ through a set of weights defined with the operator $\mathbf{L}_k^{-1}$ which approximates the inverse Laplace transform. Neurons in the second layer each have their characteristic peak time relative to the input onset $\overset{*}{\tau}_i$. **Right panel:** A response of the network to a delta-function input. Activity of only three neurons in each layer is shown. Neurons in $\mathbf{F}_{\mathbf{s}}^{(i)}$ decay exponentially with rate constants s_i and neurons $\tilde{\mathbf{f}}_{\overset{*}{\tau}}^{(i)}$ activate sequentially following the stimulus presentation. The width of the activation of each neuron scales with the peak time determined by the corresponding $\overset{*}{\tau}$, making the memory scale-invariant.

Implementation of the JOR model

To model JOR we used the memory timeline stored in $\tilde{\mathbf{f}}$, as illustrated in Figure 3a. We ran the model for 5000 trials with each trial consisting of 7 letters. We represented letters as delta pulses spaced by 100 time steps, followed by a probe composed of two randomly selected letters. Parameters of the memory representation were as follows: $k = 8$, the number of sequentially activated units for each letter was 100, the peak time of the first unit in the sequence was at 50, and the peak time of the last unit was 2000. After the probes were presented, the model scanned sequentially (from more recent towards more distant past) the memory representation $\tilde{\mathbf{f}}$ and integrated the activation of units for each probe. Once the threshold of 0.005 was reached for one of the probes, the scanning stopped and that probe was given as a response. The model was run in a noise-free case and with added noise. In the latter case, the readout from $\tilde{\mathbf{f}}$ for each probe at every step of the scanning process was multiplied by a random amount of noise. The random amount was chosen from a uniform distribution with the lower limit of 0.15 and the upper limit of 1.85.

Implementation of the JOI model

In the JOI model, we presented a sequence of 12 letters 24 times in order to learn the associative representation stored in $\mathbf{M}$, as illustrated in Figure 3b. While we could have presented the sequence only once and learn the temporal associations, presenting the sequence multiple times helped with minimizing the edge effects. Parameters of the memory representation were the same as in the JOR model. After the sequence was learned, 5000 probe trials were presented, each containing a stop position for the sequence (e.g. letter R in Figure 1b) and two randomly selected letters from the sequence that were used as probes (e.g. letters T and G in Figure 1b). Similar to the behavioral experiment from Tiganj et al. (2020), the probe letters had a lag between 2 and 7. The scanning procedure was analogous to JOR, except that instead of integrating the activation of units in the timeline of the past $\tilde{\mathbf{f}}$ we integrated the activation of the units in the timeline of the future $\mathbf{p}$ (which is equivalent to integrating corresponding weights stored $\mathbf{M}$). Similar to the JOR model, for JOI we also used a noise-free and noise case, with the same type of noise as in JOR.[1]

Results

We quantified the results from the model in terms of RT and accuracy. We compared the results with the behavioral data from Tiganj et al. (2020). All of the results resembled those obtained in the behavioral versions of JOR and JOI. For correct trials, RT depended only on the lag to the more recent (JOR) / imminent (JOI) probe and not on the lag to the less recent/imminent probe as evidenced by the flat lines in Figure 4. This was a direct consequence of the fact that we used a serial self-terminating search type of model for both memory and prediction. Furthermore, for correct trials, the RT grew sublinearly with the lag (Figure 5). This was a result of the log-compression in the model timeline for both past and future. Finally, the accuracy generated by the model resembles the accuracy observed in the behavioral data (Figure 6). For the noise-free cases, the accuracy was 1 by construction.

Discussion

We showed that an associative model based on a compressed timeline can account for major properties of RT and accuracy in both JOR and JOI. The noise-free results of the model shown in Figure 4 and Figure 5 illustrate the theoretical property of the model to provide mathematically exact logarithmic compression of the memory representation. Adding noise to the representation increased the variability of the responses and introduced some incorrect responses, but it did

[1] Both JOR and JOI models were implemented using Python and the code is available on https://github.com/zorant/JOIR-model.

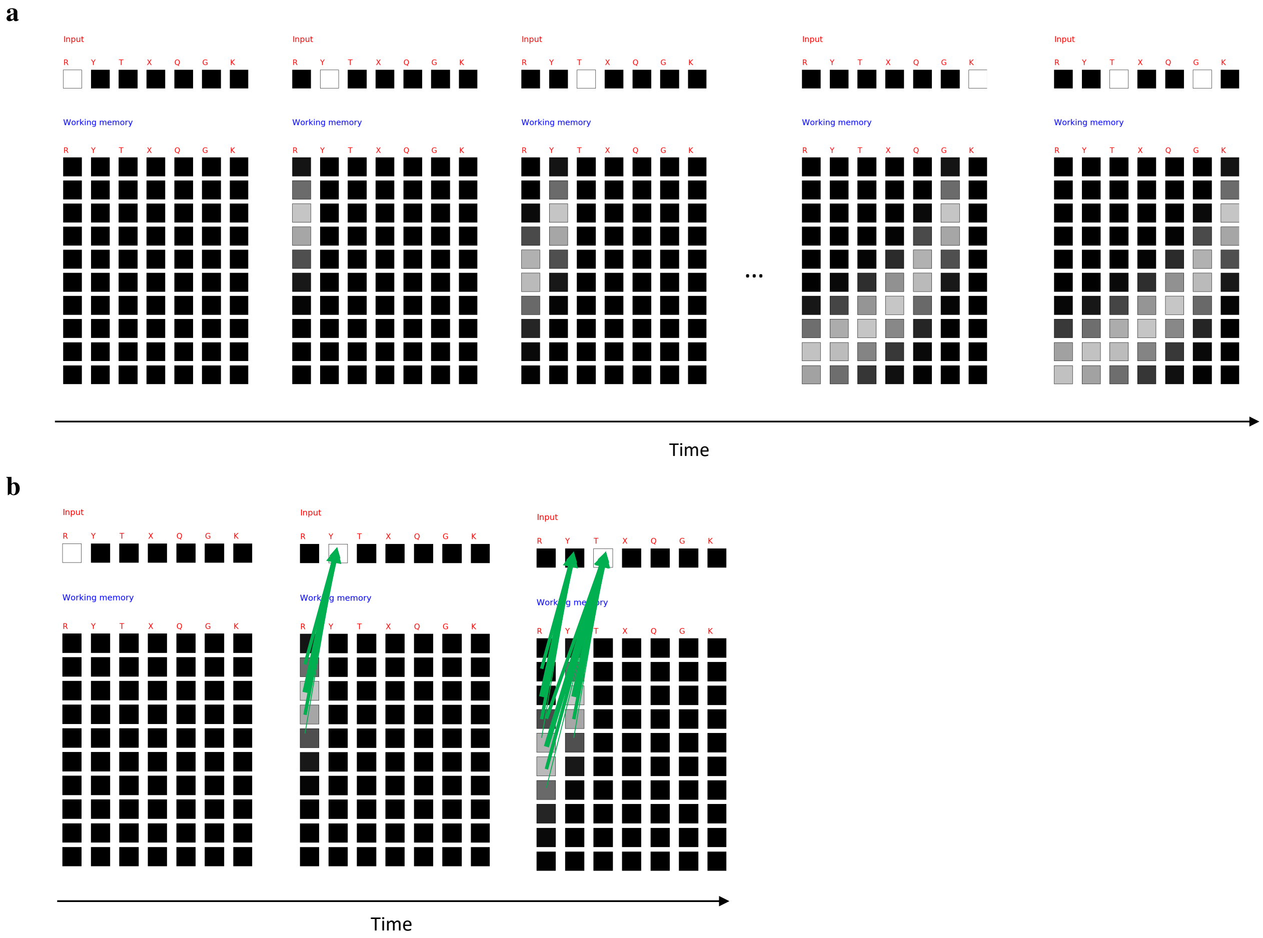

Figure 3: **Schematic of the model for JOR and JOI tasks. a.** Schematic of the memory timeline in JOR. An input sequence of 7 letters is presented sequentially. Letter presentation is marked with the white color of the letter box. Each letter has a corresponding working memory representation, which consists of sequentially activated neurons. Each box under a letter symbol represents an individual neuron in $\tilde{\mathbf{f}}^{(i)}$ where i is the index of the letter. The level of activation is coded with shades of gray such that brighter shades represent higher activation. After all 7 letters are presented, two randomly selected letters are presented as probes (in this example T and G). At that time, the memory nodes contain a compressed timeline that stores what happened when. To appreciate the compression of the timeline, notice that the white trace of activity in the memory representation is curved and gradually spreads from letters that were presented more recently to letters that were presented less recently. To provide a response, the model sequentially scans the memory representation from top to bottom and integrates the amount of activation until a threshold is reached. In the provided example, when integrating from the top to the bottom, activation for letter G will reach the threshold before the activation for T. **b.** Schematic of the associations formed between the memory timeline and the input. These associations are used to model JOI. When letter R (the first letter in the sequence) is presented, no associations are formed since the memory representation is empty. When letter Y is presented, the memory units corresponding to R that are active early in the sequence are now firing, and the associations are formed between those units and letter Y. The associations are represented with the green arrows, and the thickness of the arrows represents the strength of the associations (the weight). When letter T is presented, associations are formed with the memory units corresponding to R that activate later in the sequence and the memory units corresponding to Y that activate early in the sequence. Note that after each letter is presented, the memory weights are updated without erasing previous weights. Therefore the associations formed between Y and the early activated units of R are preserved after T is presented. After all of the letters in the sequence are presented, a wide range of associations is formed (not shown). When the probes are displayed, the memory representations of the last active letter is scanned from top to bottom. Instead of integrating activation of the memory nodes as in JOR, in JOI we integrate the strength of the associative weights. The sum of associative weights will reach the threshold for probe letter that is more imminent faster than for the probe letter that is less imminent. For instance, if after R was presented, letters Y and T appeared as probes, scanning from top to bottom through the associations formed between $\tilde{\mathbf{f}}^{(R)}$ and neurons representing $\mathbf{f}^{(Y)}$ and $\mathbf{f}^{(T)}$ would results in reaching the threshold for Y faster than the threshold for T.

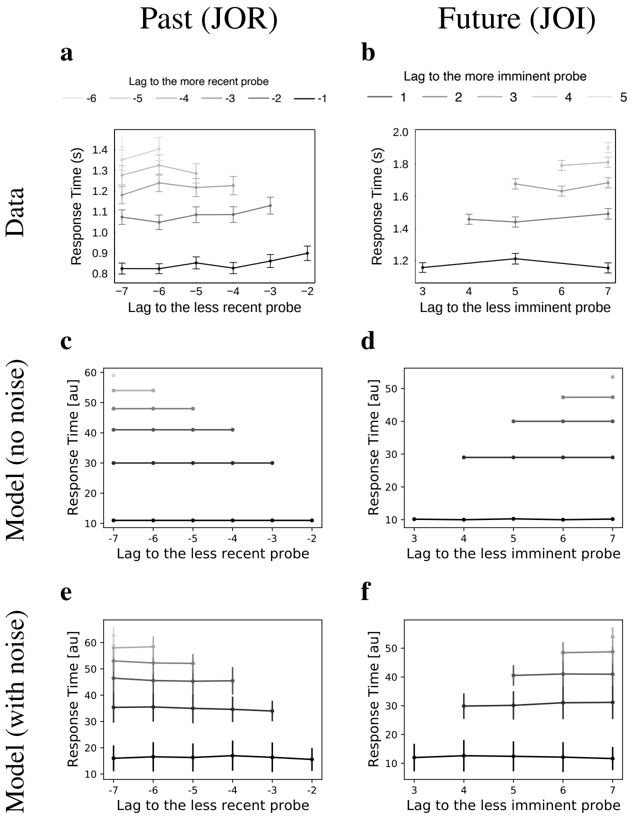

Figure 4: **RT depends on the lag of the more recent/imminent probe but not on the lag of the less recent/imminent probe for both data and the model.** This result is consistent with the hypothesis of serial self-terminating search over a timeline of the past and a timeline of the future.

not change the qualitative properties of the model. This robustness suggests that the model characterization of the data does not depend on the specific choice of the parameters. Indeed, the parameters of the model, namely *k*, the span of the memory representation and the number of neurons, do not affect the qualitative form of the results: the analytical result giving rise to the log-compressed timeline does not depend on the choice of these parameters.

The symmetry between the past and the future is remarkable for both the behavioral data and the results of the model. However, note that participants were slower in JOI than in JOR. This could be due to methodological differences between the two tasks: 1) JOI required learning of a repeated sequence, while JOR used lists presented only once, 2) the presentation rate in JOR was faster than the presentation rate in JOI. To examine the origin of this difference, future behavioral studies could explore JOR and JOI in a more matched setting. In this study, we did not try to model the impact of the presentation rate. Note that the units of time in the model do not have an interpretable meaning: the model had 100 sequentially activated units and therefore possible RTs were in

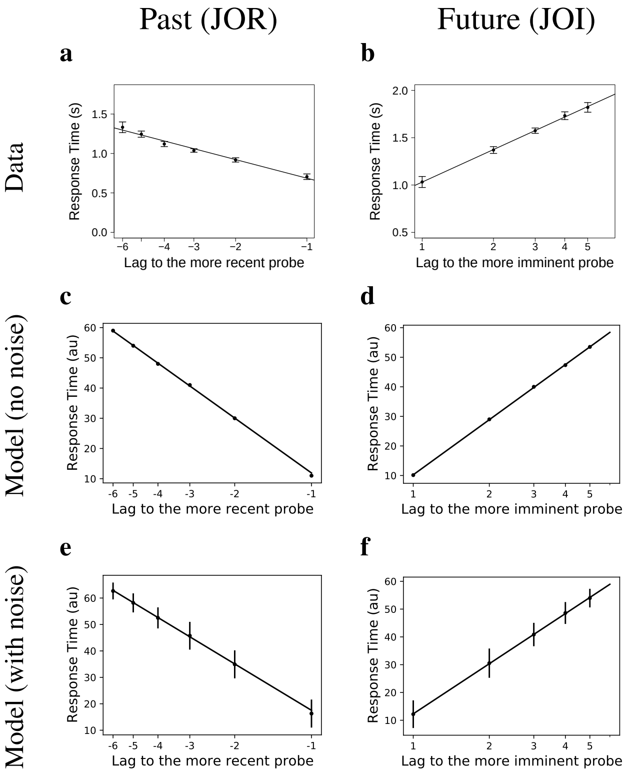

Figure 5: **The model captures the sub-linear dependence of the median RT on the lag of the more recent item indicating the compression of the memory representation.**

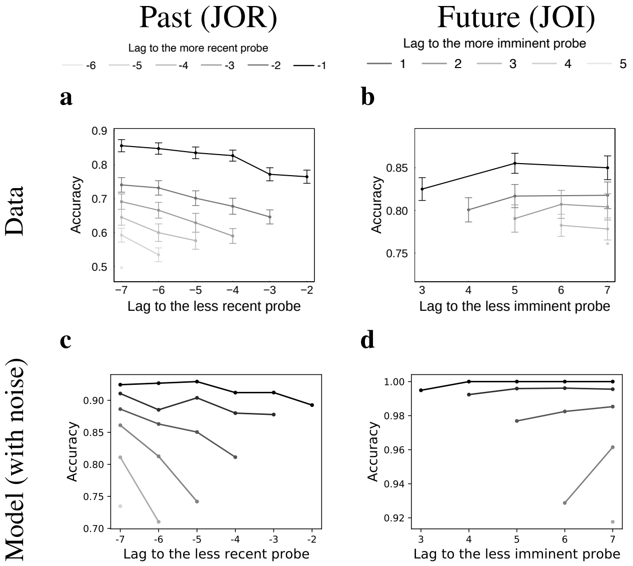

Figure 6: **The model resembles the accuracy observed in the data.**

the range from 1 to 100.

The accuracy observed with the model was higher than the accuracy in the behavioral data. While this could be compensated by increasing the amount of noise in the integrator, it is worth noting that other sources of noise likely also contribute to the behavioral results. For instance, participants may occasionally press the wrong button by a mistake. Adding such noise to the model would decrease the accuracy and it would not change the profile of responses (e.g., the functional form of relationship between RT and lag). While we did conduct the analysis for incorrect trials, the results are not shown since the amount of the behavioral data was not sufficient for a strong evidence. In incorrect trials, the model predicts that the RT will depend on the less recent/imminent item.

The proposed model is constructed on a neural-level, generating not only behavioral but also neural predictions. Existing neural data supports some of the predictions, such as sequentially activated stimulus-selective time cells (Tiganj et al., 2017; Cruzado et al., 2020) as well as exponentially decaying cells (Tsao et al., 2018; Bright et al., 2019). The JOI model also predicts the existence of sequentially activated future-time cells: neurons that activate at a characteristic time before a particular stimulus is expected to occur. Such cells have not yet been observed, and their existence would be an important neural validation of the model.

This data is rather challenging for alternative modeling approaches. In model-free learning approaches (Sutton & Barto, 1998), an agent could learn a correct answer, but there is no reason for RT to depend only on the lag of a more recent/imminent probe. On the other hand, model-based approaches (e.g. (Daw, Niv, & Dayan, 2005; Russek, Momennejad, Botvinick, Gershman, & Daw, 2017)) could learn state transitions and reapply the transition matrix until the more recent/imminent probe is reached. However, such an operation would predict a linear increase of RT with the lag, which is in contradiction with the observed behavioral data (Figure 5).

Acknowledgments

Supported by Indiana University Emerging Areas of Research (EAR) grant.

References

Bright, I. M., Meister, M. L. R., Cruzado, N. A., Tiganj, Z., Howard, M. W., & Buffalo, E. A. (2019). A temporal record of the past with a spectrum of time constants in the monkey entorhinal cortex. *bioRxiv*, 688341.

Cruzado, N. A., Tiganj, Z., Brincat, S. L., Miller, E. K., & Howard, M. W. (2020). Conjunctive representation of what and when in monkey hippocampus and lateral prefrontal cortex during an associative memory task. *Hippocampus*, *30*(12), 1332–1346.

Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, *8*(12), 1704–1711.

Hacker, M. J. (1980). Speed and accuracy of recency judgments for events in short-term memory. *Journal of Experimental Psychology: Human Learning and Memory*, *15*, 846-858.

Hockley, W. E. (1984). Analysis of response time distributions in the study of cognitive processes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*(4), 598-615.

Howard, M. W., Shankar, K. H., Aue, W., & Criss, A. H. (2015). A distributed representation of internal time. *Psychological Review*, *122*(1), 24-53.

MacDonald, C. J., Lepage, K. Q., Eden, U. T., & Eichenbaum, H. (2011). Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, *71*(4), 737-749.

McElree, B., & Dosher, B. A. (1993). Serial recovery processes in the recovery of order information. *Journal of Experimental Psychology: General*, *122*, 291-315.

Momennejad, I., & Howard, M. W. (2018). Predicting the future with multi-scale successor representations. *bioRxiv*, 449470.

Pastalkova, E., Itskov, V., Amarasingham, A., & Buzsaki, G. (2008). Internally generated cell assembly sequences in the rat hippocampus. *Science*, *321*(5894), 1322-7.

Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J., & Daw, N. D. (2017). Predictive representations can link model-based reinforcement learning to model-free mechanisms. *bioRxiv*, 083857.

Shankar, K. H., & Howard, M. W. (2012). A scale-invariant internal representation of time. *Neural Computation*, *24*(1), 134-193.

Shankar, K. H., Singh, I., & Howard, M. W. (2016). Neural mechanism to simulate a scale-invariant future. *Neural Computation*, *28*, 2594–2627.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1) (No. 1). MIT press Cambridge.

Tiganj, Z., Cromer, J. A., Roy, J. E., Miller, E. K., & Howard, M. W. (2017). Compressed timeline of recent experience in monkey lPFC. *bioRxiv*, 126219.

Tiganj, Z., Cruzado, N. A., & Howard, M. W. (2019). Towards a neural-level cognitive architecture: modeling behavior in working memory tasks with neurons. In A. Goel, C. Seifert, & C. Freksa (Eds.), *Proceedings of the 41st annual conference of the cognitive science society* (p. 1118-1123). Montreal: Cognitive Science Society.

Tiganj, Z., Gershman, S. J., Sederberg, P. B., & Howard, M. W. (2019). Estimating scale-invariant future in continuous time. *Neural Computation*, *31*(4), 681–709.

Tiganj, Z., Kim, J., Jung, M. W., & Howard, M. W. (2015). Temporal coding across scales in the rodent mPFC. *Cerebral Cortex*.

Tiganj, Z., Singh, I., Esfahani, Z., & Howard, M. W. (2020). Scanning a compressed ordered representation of the future. *bioRxiv*.

Tsao, A., Sugar, J., Lu, L., Wang, C., Knierim, J. J., Moser, M.-B., & Moser, E. I. (2018). Integrating time from experience in the lateral entorhinal cortex. *Nature*, *561*, 57-62.