Incorporating a cognitive model for evidence accumulation into deep reinforcement learning agents

James Mochizuki-Freeman (jmochizu@iu.edu)
Department of Computer Science
Indiana University Bloomington

Md Rysul Kabir (mdrkabir@iu.edu)
Department of Computer Science
Indiana University Bloomington

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

Abstract

Recent neuroscience studies suggest that the hippocampus encodes a low-dimensional ordered representation of evidence through sequential neural activity. Cognitive modelers have proposed a mechanism by which such sequential activity could emerge through the modulation of the decay rate of neurons with exponentially decaying firing profiles. Through a linear transformation, this representation gives rise to neurons tuned to a specific magnitude of evidence, resembling neurons recorded in the hippocampus. Here we integrated this cognitive model inside reinforcement learning agents and trained the agents to perform an evidence accumulation task designed to mimic a task used in experiments on animals. We found that the agents were able to learn the task and exhibit sequential neural activity as a function of the amount of evidence, similar to the activity reported in the hippocampus.

Keywords: Evidence accumulation; Cognitive model; Deep RL; Neural sequences

Introduction

Evidence accumulation is a central concept in cognitive psychology and neuroscience that describes the process by which the brain integrates sensory information over time to make decisions. Sequential sampling models, with the Drift Diffusion Model (Ratcliff, 1978) being a prominent example, posit that decision-making involves the accumulation of evidence towards one of several thresholds, each representing a different decision outcome. Decision is reached when the accumulated evidence crosses a pre-defined threshold, suggesting that enough information has been gathered to make a confident choice (Ratcliff & McKoon, 2008).

Neuroscience research supports the notion of evidence accumulation through findings that neural activity in areas such as the dorsolateral prefrontal cortex (DLPFC) and the posterior parietal cortex (PPC) reflects the process of accumulating evidence during decision-making tasks (Gold & Shadlen, 2007). For instance, the firing rates of neurons in these areas have been shown to increase in a manner consistent with the accumulation of evidence toward a decision threshold (Shadlen & Newsome, 2001).

A recent neuroscience study (Nieh et al., 2021) found that neurons in the hippocampus encode the amount of evidence such that different neurons are tuned to a different magnitude of evidence. As a population, neurons activated sequentially as a function of the amount of evidence, forming a low-dimensional manifold. Specifically, Nieh et al. (2021) trained mice on the "accumulating towers task" where while mice moved along a virtual track, objects (referred to as "towers") appeared on both sides of the track. When they arrived at the end of the track, to earn a reward, mice had to choose the left- or right-hand side, depending on which side had more towers (see also Morcos and Harvey (2016), Pinto et al. (2018) and Engelhard et al. (2019) all of which used the same experimental paradigm). The difference in the number of towers is an abstract latent variable that corresponds to the amount of evidence for either of the two options. Nieh et al. (2021) recorded the activity of hundreds of individual neurons from the dorsal CA1 sub-region of mice hippocampus while they performed the accumulating towers task. The results indicated the existence of cells tuned to a particular difference in the number of towers, such that a population of neurons tiled the entire *evidence* axis. This neural coding scheme based on sequences resembles coding of other variables in the hippocampus, specifically time and space through time cells (Pastalkova, Itskov, Amarasingham, & Buzsaki, 2008; MacDonald, Lepage, Eden, & Eichenbaum, 2011; Salz et al., 2016) and place cells (Bures, Fenton, Kaminsky, & Zinyuk, 1997) respectively (Eichenbaum, 2014; Howard & Eichenbaum, 2015).

Here we sought to construct artificial agents that receive pixel inputs very similar to those of the mice in the accumulating towers task and can solve the task while exhibiting neural sequences similar to those recorded in the hippocampus. We base our approach on a computational cognitive framework that proposed a unified representation for coding time, space, and sequences in the hippocampus (Howard et al., 2014). The framework uses a set of neurons that perform leaky integration of the input, each with a different time constant. Each such neuron has an impulse response that decays exponentially as a function of time. The output of the leaky integrators is transformed into a set of sequentially activated neurons

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

through a linear transformation that resembles lateral inhibition (Shankar & Howard, 2012). Importantly, if the decay rate of the leaky integrators is modulated by a time derivative of some variable, then the decay becomes an exponential function of that variable. For example, if the modulator is the time derivative of traveled distance (velocity), then the impulse response of the leaky integrators is an exponential function of distance. Applying the same linear transformation gives rise to neurons that sequentially activate as a function of distance, not time. An extension of this work (Howard, Luzardo, & Tiganj, 2018) demonstrated that modulating the decay rate by the change in the amount of evidence gives rise to neurons tuned to a particular magnitude of evidence. Previous work (Mochizuki-Freeman, Maini, & Tiganj, 2023) used this framework to train deep learning agents on a simple version of the accumulating towers task. The simple version of the task did not use a realistic visual environment but directly provided relevant features to the agent (three bits of information, including two bits to signal the occurrence of a tower on either side and one bit to signal the end of the environment).

Here we used a more realistic version of the environment presented in Mochizuki-Freeman, Kabir, Gulecha, and Tiganj (2023) that closely followed the design of the environment used in Nieh et al. (2021). The agent received realistic visual inputs that were then passed through an encoder composed of several convolutional layers and a single fully connected layer. The outputs of the encoder modulated the decay rate of leaky integrators in the evidence accumulation module, the input of which was set to a delta pulse at the beginning of each trial. The output of the leaky integrators underwent the same linear transformation described in (Howard et al., 2014; Shankar & Howard, 2012) and it was followed by another dense layer. Finally, the output of the dense layer was fed into the RL module based on the A2C architecture (Mnih et al., 2016). Critically, unlike in previous work (Mochizuki-Freeman, Maini, & Tiganj, 2023), the agent had to learn relevant features from the realistic environment, such that the error signal was backpropagated through the evidence accumulation module, leading to an adjustment of the weights in the encoder to select the appropriate features.

Methods

Accumulating towers task

We used the accumulating towers task environment described in Mochizuki-Freeman, Kabir, et al. (2023) (see also Lee, Engelhard, Witten, and Daw (2022) for another implementation of the same task). The environment closely mimics the virtual reality environment used in mice recordings (Nieh et al., 2021; Pinto et al., 2018). As described in Mochizuki-Freeman, Kabir, et al. (2023), the environment receives actions (*forward*, *left*, or *right*) and outputs pixel observations (Fig. 1). It has two 10.5 cm-wide arms at the end of a 200cm long straight track (Fig. 2). The agent must go through a confined path that is bounded by walls that are 10 cm apart. At the end of the track, the agent selects one of the two arms. Different numbers of towers are placed along each of the two sides with a minimum gap of 7 cm between any two towers. At each trial, the number of towers on each side is chosen randomly, between 1 and 15. The towers only appear when they are within 5 cm of the agent and disappear as the agent passes them. The towers have a 1 cm width, while the entire track has a 6 cm height. The *forward* action moves the agent down the track at a fixed speed of 1cm/step. The *left* and *right* actions change the agent's alignment in 7.5° increments to a max of $\pm 15^\circ$ along the straight portion of the maze and without a bound in the arms. Once the agent reaches the end of the track, it receives a reward of either 10 if it made a correct choice or 0 otherwise. It also receives a reward of -0.1 for attempting to turn beyond 15° in either direction when in the main track or for making contact with the back wall.

Neural-level cognitive model for evidence accumulation

We use a neural-level model for evidence accumulation described in Mochizuki-Freeman, Maini, and Tiganj (2023) that provides a neural network implementation of the framework proposed in Howard et al. (2014, 2018). Unlike previous work, here we backpropagate the error through the entire network to make the encoder learn what to feed as an input to the evidence accumulation model. The evidence accumulation model has two layers. The first layer consists of N recurrently connected neurons with activity $\mathbf{F}$ ($\mathbf{F}$ is an N long vector, where each element represents the activity of a single neuron). The strength of the recurrent connection $\mathbf{s}$ ($\mathbf{s}$ is also an N long vector) is modulated by a time derivative of some latent variable n, $\alpha = dn/dt$:

$$\frac{d\mathbf{F}(\mathbf{s};t)}{dt} = \alpha\left(-\mathbf{s}\mathbf{F}(\mathbf{s};t) + f(t)\right). \tag{1}$$

$f(t)$ is set to 1 at the beginning of each trial for a single time step and then set to 0 for the rest of the trial. By reorganizing terms in the above equation and applying the chain rule we can rewrite the equation as a function of n, instead of t:

$$\frac{d\mathbf{F}(\mathbf{s};n)}{dn} = -\mathbf{s}\mathbf{F}(\mathbf{s};n) + f(n). \tag{2}$$

If the latent variable n equals time t, then $\mathbf{F}$ would decay exponentially with rate constants $\mathbf{s}$ as a function of time (Fig. 4a). If the latent variable n is count (e.g., the number of observed towers), then $\alpha = 1$ every time a new tower is observed and 0 otherwise, resulting in $\mathbf{F}$ exponentially decaying with rate constants $\mathbf{s}$ as a function of count (Fig. 4b). For the accumulating towers task we expect the network to learn that the latent variable n should be the difference in the number of towers on the left and the right side of the wall. To make it possible for the network to learn this, we feed the output of the encoder directly to n.

Earlier work (Howard et al., 2014) has described Eq. 2 as an approximation of the Laplace transform of $f(n)$ with discrete and real s (rather than continuous and complex, as in the

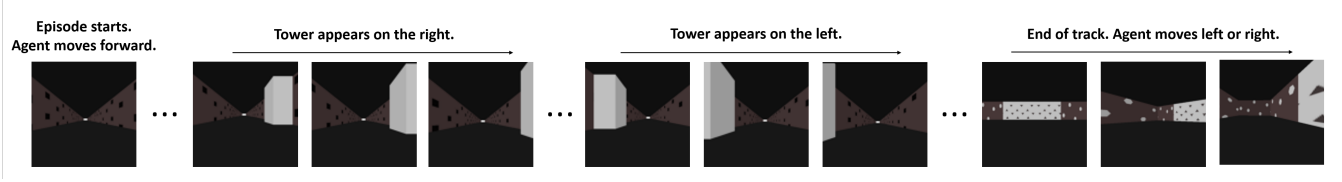

Figure 1: The accumulating towers task. In each trial, the agent moves down a narrow corridor and observes "towers" (white objects) on the left- and right-hand sides of the wall. To obtain the reward, the agent needs to turn left or right at the end of the corridor, depending on which side has more towers. Similar to the task performed by mice, the towers only become visible as the agent approaches them and then disappear shortly after. The walls of the environment have a textured pattern to provide optic flow to the agent. Adapted from (Mochizuki-Freeman, Kabir, et al., 2023).

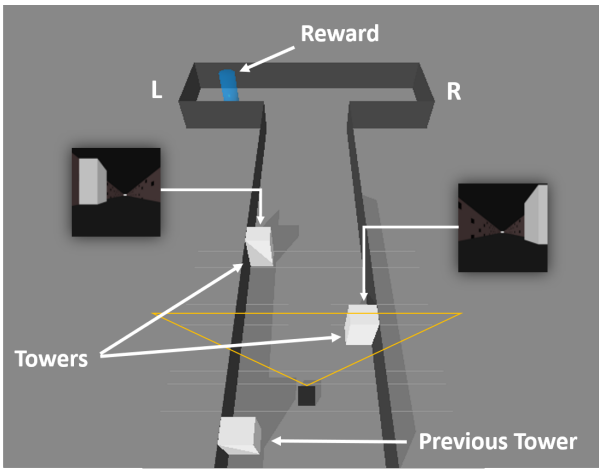

Figure 2: Schematic of the accumulating towers environment showing two inserts that illustrate the agent's visual input. Adapted from (Mochizuki-Freeman, Kabir, et al., 2023).

regular Laplace transform). Inverting the Laplace transform reconstructs the input as a function of the internal variable n^*, which corresponds to n. The inverse, which we denote as $\tilde{\mathbf{f}}(\mathbf{n}^*;t)$ can be computed using the Post inversion formula (Post, 1930):

$$\tilde{\mathbf{f}}(\mathbf{n}^*;n) = \mathbf{L}_k^{-1}\mathbf{F}(\mathbf{s};n) = \frac{(-1)^k}{k!}\mathbf{s}^{k+1}\frac{d^k}{ds^k}\mathbf{F}(\mathbf{s};n), \qquad (3)$$

where $\mathbf{n}^* := k/\mathbf{s}$ and $k \to \infty$. The reconstruction gives rise to units tuned to a particular n (bottom row in Fig. 4). By solving $\partial\tilde{\mathbf{f}}_{\mathbf{n}^*;n}/\partial n = 0$ we see that $\tilde{\mathbf{f}}(\mathbf{n}^*;n)$ peaks at $n^* = n$.

As in Mochizuki-Freeman, Maini, and Tiganj (2023), for a neural network implementation, we discretize the Laplace and inverse Laplace transform for both $\mathbf{s}$ and t. We write a discrete-time approximation of Eq. (1) as an RNN with a diagonal connectivity matrix and a linear activation function:

$$\mathbf{F}_{s;t} = \mathbf{W}\mathbf{F}_{s;t-1} + f_t, \qquad (4)$$

where $\mathbf{W} = \mathrm{diag}(e^{-\alpha(t)s\Delta t})$. A discrete approximation of the inverse Laplace transform, $\tilde{\mathbf{f}}_{n^*;t}$, can be implemented by multiplying $\mathbf{F}_{s;t}$ with a derivative matrix $\mathbf{L}_k^{-1}$ computed for some finite value of k.

Agent architecture

We constructed deep RL agents based on the A2C architecture (Mnih et al., 2016) (Fig. 3). The agents consisted of three main modules: a convolutional encoder to reduce an input image to a latent vector, a neural-level cognitive model for evidence accumulation described in the previous section, and actor and critic networks to generate action and value predictions.

We used the same encoder layer as Mochizuki-Freeman, Kabir, et al. (2023). At each time step, the agents received as input 60×60 grayscale pixel values from the environment. Those images were fed into three successive convolutional layers: 64 kernels of size 8×8 (stride 2), 32 kernels of size 2×2 (stride 1), and 64 kernels of size 3×3 (stride 2). This reduced the input to tensors of size $27 \times 27 \times 64$, then $26 \times 26 \times 32$, and then $12 \times 12 \times 64$, respectively. The resulting tensor was flattened into a 9216-element vector and passed through a fully connected layer, which reduced the 9216 intermediate units to 32 output features. All layers in the encoder had ReLU activation functions.

The convolutional encoder was followed by the evidence accumulation network consisting of 32 $\tilde{\mathbf{f}}$ modules, with $N = 20$ (total of 640 $\tilde{\mathbf{f}}$ neurons). Each encoder output was fed into the modulatory input of one evidence accumulation module. Parameter k was set to 8 and n^* consisted of N log-spaced values between 0.1 and 20.

To evaluate the contribution of the evidence accumulation network, we also conducted experiments where it was replaced by commonly used recurrent neural networks: a simple RNN, GRU, and LSTM, each consisting of 640 neurons to match the number of neurons in $\tilde{\mathbf{f}}$. To better understand the computational role of sequentially activated cells in comparison to exponentially decaying cells, we also conducted experiments on an evidence accumulation network that had only $\mathbf{F}$ neurons, without $\tilde{\mathbf{f}}$ neurons.

The output of the evidence accumulation layer was passed to an actor network and a critic network, which produced action logits and state value estimates, respectively. The actor and the critic network each consisted of two-layers: a fully connected layer of 640 neurons followed by a fully connected output layer of 3 (actor head) or 1 (critic head) neurons.

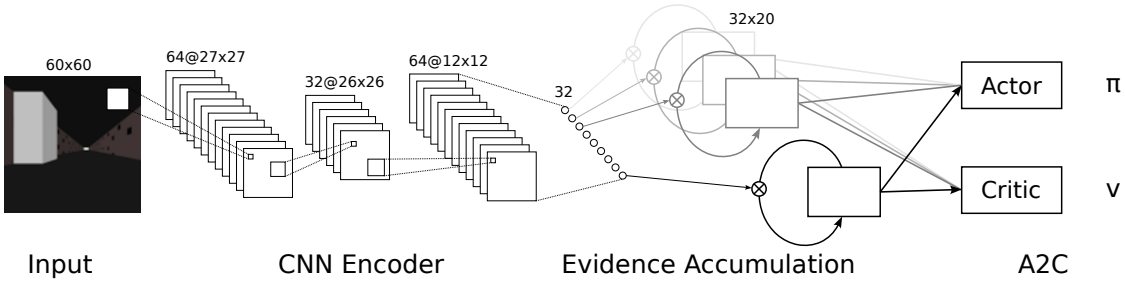

Figure 3: Agent architecture. Pixel inputs are fed into a CNN encoder followed by a neural-level cognitive model for evidence accumulation. The output of the evidence accumulator is fed into an A2C-based RL agent.

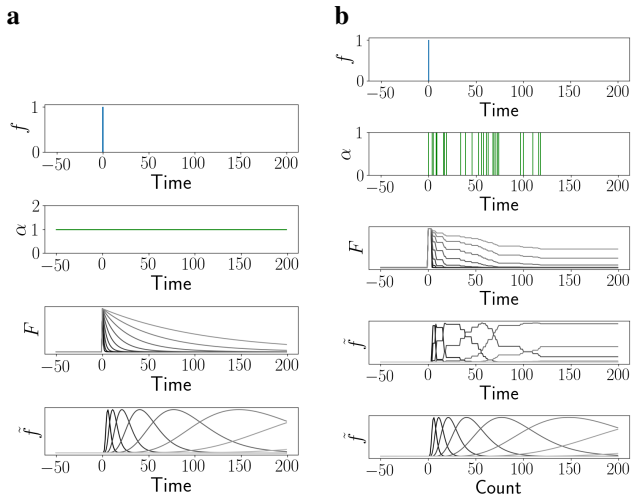

Figure 4: Example of the Laplace and inverse Laplace transform with and without modulatory input. (a) In the absence of modulatory input, the impulse response of the Laplace transform decays exponentially with decay rate s. The impulse response of the inverse Laplace transform has a unimodal shape. Note that if time t was shown on the log-scale, the unimodal curves would be equally wide and equidistant providing a log-compressed representation consistent with Weber's law. (b) α modulates the decay rate of units in $\mathbf{F}$ and it is proportional to the change in the count. This makes units in $\tilde{\mathbf{f}}$ develop unimodal basis functions that are tuned to count rather than to time and peak at n^*.

Training

During training, we explored five different learning rates (0.001, 0.0005, 0.0001, 0.00005, and 0.00001) for the Adam optimizer, and three different entropy bonus coefficients (0.0, 0.0001, and 0.00001). We ran each agent configuration four times for 12 million environment steps. For each training step, we fed into the agent 5120 total environment steps, collected from 20 environments running in parallel for 256 time steps each, and then performed a backpropagation pass and stepped the optimizer.

	Learning Rate	Entropy
$\tilde{\mathbf{f}}$	0.0001	0
$\mathbf{F}$	0.0001	0.0001
GRU	0.00005	0.00001
LSTM	0.00005	0.0001
RNN	0.00001	0.00001

Table 1: Hyperparameters of the top-performing agents for each of the different agent architectures.

Results

We trained and evaluated five different groups of agents on the accumulating towers task. Two groups were based on evidence accumulation model: one group included the full model with Laplace and inverse Laplace transform ($\tilde{\mathbf{f}}$), and the other included only the Laplace transform ($\mathbf{F}$). We also compared three groups of agents based on existing recurrent architectures: a GRU, LSTM, and simple RNN. For each of the five models, we performed 100 validation runs every 1000 episodes. At the end of the training, we selected the hyperparameter configuration for each model that had the highest average validation reward. The top-performing hyperparameters for each model are summarized in Table 1.

Agents based on the cognitive model learned the task as fast as the GRU-based agents despite having much fewer parameters

The $\tilde{\mathbf{f}}$ agents learned the task and converged to a reward value of 10 at a similar number of training steps as the GRU and LSTM agents (Fig 5 and Fig. 6). This indicates that the sequential map-like representation provided by the cognitive model could be used by actor and critic networks to successfully learn the task, even without the recurrent network containing any trainable parameters. On the other hand, the RNN and $\mathbf{F}$ agents did not learn the task within 12M environment steps highlighting the importance of the inverse transform which gives rise to the sequential representation.

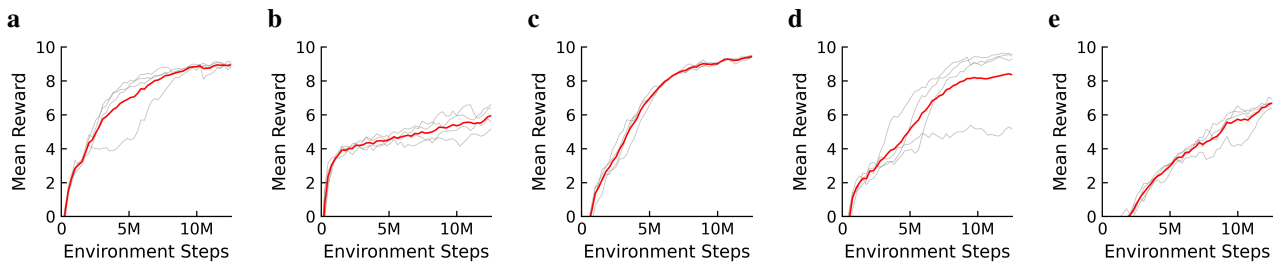

Figure 5: Agent performance on the accumulating towers task. Each gray line represents the performance of a single agent over the course of training. The red line represents the mean performance of the four agents. The maximum mean reward was 10. Agent architecture: (a) $\tilde{\mathbf{f}}$, (b) **F**, (c) GRU, (d) LSTM, (e) RNN.

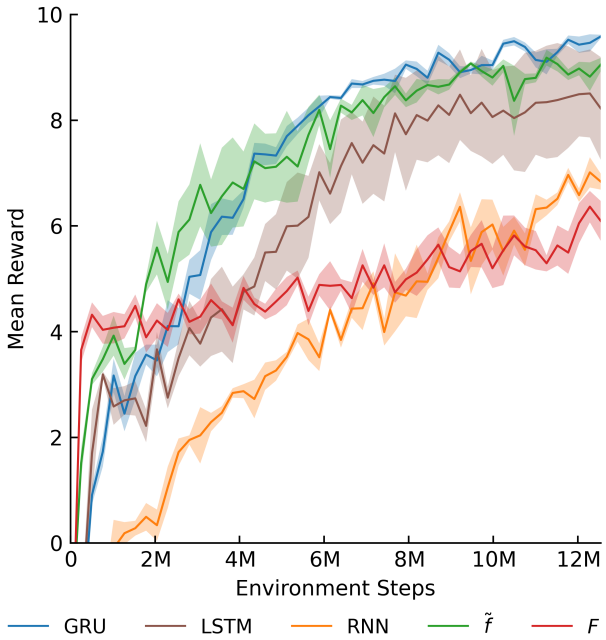

Figure 6: Agent performance on the accumulating towers task. Mean and standard deviation across the four agents.

	Encoder	Memory	RL	Total
$\tilde{\mathbf{f}}$	325,824	0	823,044	1,148,868
F	325,824	0	823,044	1,148,868
GRU	325,824	1,294,080	823,044	2,442,948
LSTM	325,824	1,725,440	823,044	2,874,308
RNN	325,824	431,360	823,044	1,580,228

Table 2: Number of parameters for different agent architectures and different components. Note that the evidence accumulation module has no trainable parameter hence $\tilde{\mathbf{f}}$ and **F** have no trainable memory parameters.

Neural activity inside the recurrent layer resembles activity in mice hippocampus

We visualized the neural activity of one representative agent for each of 5 models after 10M environment steps of training (Fig. 7). The neurons in the $\tilde{\mathbf{f}}$ agents activated sequentially as a function of evidence, resembling the activity in neural recordings from the hippocampus (Nieh et al., 2021).

Discussion

Advancements of pixel-to-action deep RL agents (Mnih et al., 2013, 2015; Silver et al., 2017) provide an opportunity for integration with neural-level cognitive models in realistic behavioral tasks. The neural activity of such agents can be compared to the neural activity in electrophysiological recordings, and performance can be compared to animal performance.

Here we integrated a neural-level evidence accumulation model (Howard et al., 2014; Mochizuki-Freeman, Maini, & Tiganj, 2023) into pixel-to-action deep RL agents. This work constitutes an important test for understanding whether a neural-level cognitive model can become a working part of a differentiable architecture. In previous work, Maini et al. (2023) demonstrated on simple toy-examples that a neural network can select basic modulatory features using the framework from Howard et al. (2014). Here we scaled this to a realistic behavioral task in a realistic visual environment.

We found that agents based on the evidence accumulation model learned as well as agents based on GRU and LSTM architectures despite having less than half of the parameters as those agents (Table 2). In particular, recurrent weights in the evidence accumulation model were not trainable, resulting in much fewer parameters than in GRU and LSTM agents. We also found that agents that included only the Laplace but not the inverse Laplace transform did not learn. The inverse transform is a simple linear mapping that resembles lateral inhibition (it implements a spatial derivative of order k). This mapping turns exponentially decaying neurons into sequentially activated ones. It appears that sequential activation put the network in a position where learning was more efficient. This is consistent with previous work on spatial navigation, where the existence of place and grid cells resulted in faster learning (Banino et al., 2018).

Previous work has shown that under particular circumstances (mnemonic demand), neural activity in deep reinforcement learning agents resembles the activity of time cells (Lin & Richards, 2021). Similarly, Banino et al. (2018) has shown the emergence of grid cells in deep reinforcement

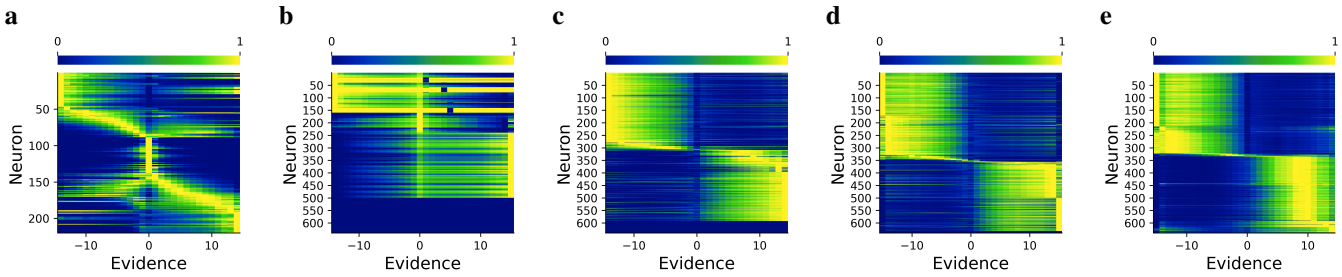

Figure 7: Neural activity of agents after 10M steps of accumulating towers task. Similar to plots in (Nieh et al., 2021; Morcos & Harvey, 2016), neurons are sorted by peak activity. Each row is normalized such that the activity ranges from 0 to 1. Agent architecture: (a) $\tilde{\mathbf{f}}$, (b) $\mathbf{F}$, (c) GRU, (d) LSTM, (e) RNN. While we had a total of four agents for each architecture, here we displayed one representative agent. The neural activities in other agents were qualitatively similar to the selected examples. For $\tilde{\mathbf{f}}$ agent, only neurons that had non-zero activity are shown (220 out of 640).

learning agents during a spatial navigation task. Here we observed the emergence of cells that activate sequentially as a function of evidence, but only for the agents based on the evidence accumulation model. Specifically, neural activity in the $\tilde{\mathbf{f}}$ layer exhibited sequential activation as a function of evidence (Fig. 7a), similar to Nieh et al. (2021). This implies that the encoder learned to increase activity in response to the change in the amount of evidence (appearance of a new tower). Importantly, if the encoder was sensitive only to the appearance of a new tower on one side of the wall, then $\tilde{\mathbf{f}}$ neurons would activate sequentially as a function of the total number of towers on that side of the wall. The fact that they activated sequentially as a function of evidence suggests that the encoder actually learned to subtract the number of towers on the two sides by appropriately modulating the recurrent weights. Specifically, the encoder learned to change the baseline activity of α every time a new tower would appear. The magnitude of the change was either positive or negative, depending on whether the tower appeared on the left or the right side of the wall. Agents based on other architectures, including $\mathbf{F}$, RNN, GRU and LSTM (Fig. 7b-e) were mainly characterized by decaying and growing neural activity as a function of the amount of evidence. This is in contrast to the neural data reported in Nieh et al. (2021).

Previous work (Mochizuki-Freeman, Maini, & Tiganj, 2023) used the same cognitive model but did not have realistic visual inputs and a trainable encoder. That work also explicitly computed the subtraction between different evidence accumulation modules by using computational properties of the Laplace domain. The subtraction was useful since the amount of evidence is the subtraction between the number of towers on the two sides. Since we used 32 evidence accumulation modules here, computing a subtraction was not computationally feasible. However, as mentioned above, the encoder learned to modulate the decay rate of the nodes in the evidence accumulator differently depending on which side of the wall the towers appeared.

Future work should investigate how to better utilize the structured representation in $\tilde{\mathbf{f}}$. In the present work, while $\tilde{\mathbf{f}}$ consisted of a structured representation where units activated sequentially as a function of the amount of evidence, it was fed into fully connected layers that are part of the actor-critic architecture. These fully connected layers are initialized with random weights, therefore destroying the structure of the representation. While the agents still benefited from having $\tilde{\mathbf{f}}$, we hypothesize that networks designed to preserve the structure will lead to better performance (e.g., faster learning). An example of such a network is a convolutional neural network, which takes advantage of the spatial structure. In general, networks capable of deploying attention to parts of $\tilde{\mathbf{f}}$ or navigating $\tilde{\mathbf{f}}$ as searchable space could lead to more human-like evidence accumulation and decision-making.

Acknowledgments

We gratefully acknowledge support from the National Institutes of Health's National Institute on Aging, grant 5R01AG076198-02. This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., ... others (2018). Vector-based navigation using grid-like representations in artificial agents. *Nature*, *557*(7705), 429–433.

Bures, J., Fenton, A., Kaminsky, Y., & Zinyuk, L. (1997). Place cells and place navigation. *Proceedings of the National Academy of Sciences*, *94*(1), 343–350.

Eichenbaum, H. (2014). Time cells in the hippocampus: a new dimension for mapping memories. *Nature Reviews Neuroscience*, *15*(11), 732–744.

Engelhard, B., Finkelstein, J., Cox, J., Fleming, W., Jang, H. J., Ornelas, S., ... others (2019). Specialized coding of sensory, motor and cognitive variables in vta dopamine neurons. *Nature*, *570*(7762), 509–513.

Gold, J. I., & Shadlen, M. N. (2007). The neural basis of decision making. *Annual Review of Neuroscience*, *30*, 535–574.

Howard, M. W., & Eichenbaum, H. (2015). Time and space in the hippocampus. *Brain research*, *1621*, 345–354.

Howard, M. W., Luzardo, A., & Tiganj, Z. (2018). Evidence accumulation in a laplace domain decision space. *Computational brain & behavior*, *1*(3), 237–251.

Howard, M. W., MacDonald, C. J., Tiganj, Z., Shankar, K. H., Du, Q., Hasselmo, M. E., & Eichenbaum, H. (2014). A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience*, *34*(13), 4692–4707.

Lee, R. S., Engelhard, B., Witten, I. B., & Daw, N. D. (2022). A vector reward prediction error model explains dopaminergic heterogeneity. *bioRxiv*, 2022–02.

Lin, D., & Richards, B. A. (2021). Time cell encoding in deep reinforcement learning agents depends on mnemonic demands. *bioRxiv*.

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.

Maini, S. S., Mochizuki-Freeman, J., Indi, C. S., Jacques, B. G., Sederberg, P. B., Howard, M. W., & Tiganj, Z. (2023). Representing latent dimensions using compressed number lines. In *2023 international joint conference on neural networks (ijcnn)* (pp. 1–10).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mochizuki-Freeman, J., Kabir, M. R., Gulecha, M., & Tiganj, Z. (2023). Geometry of abstract learned knowledge in deep rl agents. In *Neurips 2023 workshop on symmetry and geometry in neural representations.*

Mochizuki-Freeman, J., Maini, S. S., & Tiganj, Z. (2023). Characterizing neural activity in cognitively inspired rl agents during an evidence accumulation task. In *2023 international joint conference on neural networks (ijcnn)* (pp. 01–09).

Morcos, A. S., & Harvey, C. D. (2016). History-dependent variability in population dynamics during evidence accumulation in cortex. *Nature neuroscience*, *19*(12), 1672–1681.

Nieh, E. H., Schottdorf, M., Freeman, N. W., Low, R. J., Lewallen, S., Koay, S. A., ... Tank, D. W. (2021). Geometry of abstract learned knowledge in the hippocampus. *Nature*, 1–5.

Pastalkova, E., Itskov, V., Amarasingham, A., & Buzsaki, G. (2008). Internally generated cell assembly sequences in the rat hippocampus. *Science*, *321*(5894), 1322–1327.

Pinto, L., Koay, S. A., Engelhard, B., Yoon, A. M., Deverett, B., Thiberge, S. Y., ... Brody, C. D. (2018). An accumulation-of-evidence task using visual pulses for mice navigating in virtual reality. *Frontiers in behavioral neuroscience*, *12*, 36.

Post, E. (1930). Generalized differentiation. *Transactions of the American Mathematical Society*, *32*, 723-781.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873–922.

Salz, D. M., Tiganj, Z., Khasnabish, S., Kohley, A., Sheehan, D., Howard, M. W., & Eichenbaum, H. (2016). Time cells in hippocampal area ca3. *Journal of Neuroscience*, *36*(28), 7476–7484.

Shadlen, M. N., & Newsome, W. T. (2001). Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. *Journal of Neurophysiology*, *86*(4), 1916–1936.

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

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... others (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354–359.