Modeling the Role of Working Memory and Episodic Memory in Behavioral Tasks

Eric A. Zilli[1,2]* and Michael E. Hasselmo[1,2]

ABSTRACT: The mechanisms of goal-directed behavior have been studied using reinforcement learning theory, but these theoretical techniques have not often been used to address the role of memory systems in performing behavioral tasks. This work addresses this shortcoming by providing a way in which working memory (WM) and episodic memory may be included in the reinforcement learning framework, then simulating the successful acquisition and performance of six behavioral tasks, drawn from or inspired by the rat experimental literature, that require WM or episodic memory for correct performance. With no delay imposed during the tasks, simulations with WM can solve all of the tasks at above the chance level. When a delay is imposed, simulations with both episodic memory and WM can solve all of the tasks except a disambiguation of odor sequences task. © 2007 Wiley-Liss, Inc.

KEY WORDS: delayed nonmatch; hippocampus; reinforcement learning; sequence disambiguation; spatial alternation

INTRODUCTION

Reinforcement learning theory (Sutton and Barto, 1998) provides a powerful theoretical framework for describing problems in which an agent (an animal, robot, etc.) interacts with its environment by selecting actions and receives a scalar reward signal after each action is taken. Recently, much attention has been paid to the possibility that a similar system is used in the brains of animals, especially because midbrain dopaminergic neuron firing appears to resemble the error signal used in this framework (Schultz et al., 1997; Montague et al., 2004; but see Morris et al., 2006; Redgrave and Gurney, 2006). Neural network models of reinforcement learning that do not rely on dopamine have also been proposed (Hasselmo, 2005; Koene and Hasselmo, 2005; Zilli and Hasselmo, 2005).

Traditional approaches to reinforcement learning are limited by the requirement that tasks or environments have the Markov property. Roughly, this requirement means that the optimal action to take in a particular observed state must depend only on that state and not on the recent history of the agent. The theory thus does not easily apply to many of the common behavioral tasks that require an animal to make a decision based on memory of a past state, such as delayed matching to sample or spatial alternation. Lin and Mitchell (1992) overcame this limitation by using order-*n* Markov chains, in which an agent's state becomes the set of its *n* most recent observations. McCallum (1995) improved on this idea using his utile-distinction learning approach by which an agent could learn to include aspects of a state from up to a fixed number of steps in the past into its current state. In this way, an agent could act more flexibly by incorporating only the most relevant of the most recent *n* observations instead of all of them. A drawback to this approach is that it still does not overcome the inflexibility that arises from referring to observations from a fixed number of states in the past.

Part of the flexibility that animals show during behavior arises from their access to multiple memory systems that provide a variety of information. Two commonly studied memory systems are working memory (WM) (Baddeley, 1986) and episodic memory (Tulving, 2002). WM has been defined as memory "active and relevant only for a short period of time" (Goldman-Rakic, 1995), whereas episodic memory is a longer lasting memory that allows one to recall and re-experience personal events (Tulving, 2002). A given task might be solvable using more than one distinct system (Redish, 2001), which complicates understanding how an animal may perform and which memory systems will be used. Here, we introduce simple, abstract forms of these two memory systems in the reinforcement learning framework and simulate common behavioral tasks from experimental work on memory systems.

First, we describe the ideas and implementation of abstract versions of WM and episodic memory. Then, we demonstrate the memory systems on a few common tasks, some of which can be solved with either WM or episodic memory and some by only one of the two. The tasks are all presented in delayed and nondelayed conditions, in which the simulated effects of the delay prevent the agent from using a WM strategy, thus demonstrating whether episodic memory is capable of solving the task.

The primary insight that allowed these tasks to be modeled in this framework was that the memory systems of an agent can be treated as part of the environ-

[1] **Program in Neuroscience, Center for Memory and Brain, Boston University, Boston, Massachusetts;** [2] **Department of Psychology, Center for Memory and Brain, Boston University, Boston, Massachusetts**

Grant sponsor: NIH; Grant numbers: Silvio O. Conte Center MH71702, MH60013, DA16454NSF Science of Learning Center; Grant numbers: SBE 0354378

*Correspondence to: Eric A. Zilli, Program in Neuroscience, Department of Psychology, Center for Memory and Brain, Boston University, 2 Cummington St., Boston, MA 02215. E-mail: zilli@bu.edu

Accepted for publication 1 October 2007
DOI 10.1002/hipo.20382
Published online in Wiley InterScience (www.interscience.wiley.com).

ment instead of part of the agent, allowing the agent to take actions upon its memory systems in the same way it takes actions on its physical environment. Environment in this context does not refer to the common usage of the word, but rather to the technical definition from reinforcement learning in which the environment can be defined as anything that provides state input to the agent and changes its state in response to the agent's actions. Although the memory systems are environment, we will still refer to "the agent's WM," for instance, where the possessive is meant to indicate not containment, but rather possession.

MATERIALS AND METHODS

Reinforcement Learning

Recent work from this laboratory has shown that reinforcement learning type systems can be implemented in neural circuits (Hasselmo, 2005; Koene and Hasselmo, 2005; Zilli and Hasselmo, 2005). For the purposes of this work, however, we used the more common, abstract implementation of the system, because our focus here was on demonstrating that use of memory systems can be treated as an active, learned process. These simulations were written in C++, and the source code is available upon request.

We used a tabular, actor-critic temporal difference learning architecture with ε-greedy exploration (Sutton and Barto, 1998). In this architecture, the agent maintains a table of state values *V*, where *V*(*s*) is the agent's current estimation of the expected, temporally discounted reward that will follow state *s* (thus more temporally distant rewards are considered as having less value than immediate rewards of the same magnitude). The agent also maintains a table of action values *Q*, where *Q*(*s*,*a*) is the value of taking action *a* in state *s*.

The agent followed a simple policy in which it took the action with the highest action value in its current state with ties broken randomly, except, with probability $0 \leq \varepsilon \leq 1$ the agent selected an action completely at random. All state and action values were initialized to 0 at the beginning of a simulation, and so the agent always selected a random action the first time it entered a novel state. This is the *actor* half of the architecture.

When the agent takes action *a* in state *s* and finds itself in state *s*′ having received a scalar reward *r*, the TD error δ is computed by the *critic* half of the architecture:

$$\delta = r + \gamma V(s') - V(s) \tag{1}$$

where $0 \leq \gamma \leq 1$ is a temporal discounting factor (smaller values of γ cause the agent to consider immediate rewards to be more valuable than distant rewards). This is the difference between the value of the old state (*s*) and the sum of the reward signal (*r*) and the discounted value of the new state (*s*′) and is used to update the old state's value and the action value for the action just taken.

$$V(s) \leftarrow V(s) + \alpha\delta$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\delta$$

where α is a learning rate parameter. If $\delta > 0$ then *V*(*s*) was less than the combined reward and discounted value of the new state, and so *V*(*s*) should be increased to provide a better estimate. Similarly, if $\delta < 0$, then the agent's estimated value of state *s* was found to be too high, and so *V*(*s*) should be decreased. No learning occurs if the predicted state value exactly matches the sum of the reward and the discounted value of the new state.

To speed up the agent's learning of the tasks, replacing eligibility traces were used (Singh and Sutton, 1996; Sutton and Barto, 1998). With eligibility traces, multiple state and action values are updated on each step with the update scaled proportional to the amount of time because the agent last visited the state and took the action. Now receipt of a reward while entering the current state affects values not only at the previous state but also at the most recently visited states. Each time the agent leaves state *s* taking action *a*, we assign $E(s,a) \leftarrow 1$ and for all other states *s*′ we assign $E(s',a) \leftarrow \lambda E(s',a)$, where λ is a decay coefficient. The learning equations become

$$V(s) \leftarrow V(s) + \alpha\delta\, E(s,a) \tag{2}$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\delta\, E(s,a) \tag{3}$$

for each pair of *s* and *a*. In the simulations, *E*(*s*,*a*) was set to 0 if it was below a threshold value θ.

In all simulations presented here, α was set to 0.01, γ was set to 0.95, ε was set to 0.01, λ was set to 0.25, and θ was set to 0.01, so that at most four state-action pairs had nonzero eligibility trace strengths at any time.

States and Actions

We assume a factored representation of state $S = S_1 \times S_2 \times \cdots \times S_n$, so elements of *S* are *n*-tuples $\{(x_1, x_2, \ldots, x_n) | x_1 \in S_1, x_2 \in S_2, \ldots\}$. In the simulations below, there were usually two or three state elements: the location state S_L (numbered square in a gridworld) and the WM and episodic memory states S_{WM} and S_{EP} (described below). We thus had $S = S_L \times S_{\mathrm{WM}}$ for WM simulations and $S = S_L \times S_{\mathrm{EP}} \times S_{\mathrm{WM}}$ for episodic memory simulations, in which, for example, each triple is of the form (location, episodic memory contents, WM contents), and each such double or triple has its own set of action values. As noted in the results, in a few tasks, an extra state element was used to represent sensory input: olfactory input in the odor sequence disambiguation task, visual input for barriers in the nonmatch to position task, and auditory input in the tone-cued alternation task.

We include two types of actions: motor actions (north, south, east, west), which change the location state S_L, and we

TABLE 1.

Summary of States and Actions in Working Memory Agent Simulations

States	Actions	State transition
S_L, a sensory state	Motor action (e.g., N/S/E/W movement, lever press, and dig for reward in cup of sand)	$S_L \leftarrow$ sensory state S_{WM} unchanged
S_{WM}, a sensory state or empty	Add to buffer	S_L unchanged $S_{WM} \leftarrow S_L$

introduce two new types of memory actions that change S_{WM} and S_{EP}.

Notice that, because the agent receives state information reflecting memory and can take actions affecting memory states, from a physiological point of view, we are effectively including many parts of cortex as a part of the agent's environment and not of the agent itself. We leave the specific location of the agent undefined, but it might be thought of as basal ganglia or medial prefrontal and orbitofrontal cortices, which are thought to be involved in learning and decision making (Houk et al., 1995; Schoenbaum and Setlow, 2001; Krawczyk, 2002; O'Reilly and Frank, 2006). Our present results, however, do not rely on any particular anatomical locus for the agent. Sutton and Barto (1998) suggested that muscles and sensory organs, for example, should be considered environment and not agent. We are simply further restricting the agent to represent roughly the highest levels of the cortical hierarchy, which can take actions mediated by neurophysiological interactions with other regions.

Working Memory

The traditional definition of WM is the ability to maintain a representation of a stimulus online and to be able to manipulate it mentally (Baddeley, 1986), but for our present purposes we will consider only the former aspect. The first action we introduce is an action that sets S_{WM} to the current sensory state, keeping that state in memory until it is overwritten when the next memory action is taken. Until this action is initially taken or if memory is cleared (e.g., to simulate imposition of a long delay during a task), an extra state representing empty memory is used. Therefore, $|S_{WM}| = |S_L| + 1$, the number of possible WM states is one more than the number of possible location states. This functionality is summarized in Table 1. In the future work, this type of action could be extended in a few obvious ways. First, one could allow multiple WM "buffers," allowing a separate action for each buffer, for example, states S_{WM1} and S_{WM2} with one action to store a sensory state in S_{WM1} until that storage action is taken again (and replaces S_{WM1}) and another action to store a sensory state in S_{WM2} until that storage action is taken again. Another extension would be to allow separate actions to buffer the value of a specific state element. For example, one could have separate WM buffers for visuospatial state and for auditory state, as in the WM model of Baddeley and Hitch (1974).

This idea is similar to the more biological neural network model of gating WM updating proposed by O'Reilly and Frank (2006). In both mechanisms, the agent learns the appropriate times in a task at which to buffer stimuli into WM by increasing the action value of the WM action (or gating action in the O'Reilly and Frank, 2006 model) at the proper states. The concept also overlaps with the idea of options as described by Sutton et al. (1999), which are temporally extended actions that change the agent's policy until the option terminates, essentially providing the agent a continuing memory of the ongoing option. Sutton et al. (1999) defined an option formally as a triple $\langle I, \pi, \beta \rangle$, where I is the set of states where the option is available as an action, π is the policy to be followed for the duration of the action (i.e., a set of Q-values), and $\beta(s)$ is a function giving the probability for each state s that the option will terminate in that state. Thus, our WM concept is roughly equivalent to a set of options, each with a single initiation state, reflecting the specific state element to be buffered in WM (i.e., in a two-by-two environment, there would be four options or four possible states to hold in WM). Because the model uses a factored state representation in which WM content is one element, each possible item in WM effectively defines a separate policy. With this interpretation, in our delayed tasks, the probability of terminating at the location in which the delay is enforced is 1 and is 0 elsewhere. However, an item in WM can also be overwritten at any state by a new one, at which point the old item is lost. Options, on the other hand, may be nested, so that by taking one option, then taking another, when the second option terminates the first one may still apply. Hence the concept of an option and our WM mechanism are similar, but distinct in certain details.

Episodic Memory

The next two actions we introduce are episodic memory actions. First, however, we describe an abstract episodic memory system to motivate the new actions. The two important characteristics of this memory system are (1) it is content addressable and (2) it is temporally indexed (this description is somewhat different from the phenomenological characterization used by Tulving, 2002). Content addressability means that a memory can be elicited from a cue that is contained as part of the memory (Marr, 1971; e.g., given the cue word "breakfast," one can recall a specific memory containing the concept of breakfast). The term "temporally indexed" means that once a memory is retrieved, one can "replay" the memory from the retrieved point as if it were a movie. In contrast, semantic memory is effectively content addressable memory that lacks a temporal index: one can retrieve specific facts given associated facts, but there is no sense of temporal progression (e.g., given USA and capital one might retrieve Washington, D. C., but the concept of "the next moment in time" given such a memory is undefined).

We now describe a simple episodic-like memory system designed to model these two characteristics. A specific mathematical implementation and a biological interpretation for it can be found in Appendix.

The model maintains a list of the *n* most recently visited states in order. After each action, the agent's previous state is added to the end of the list, and the oldest state in the list is removed (however, no state is removed for the first *n* steps). This list is the complete episodic memory of the agent. The agent's episodic memory state S_{EP} will contain either one state from this list or an additional state representing "nothing recalled." Each time the agent takes a spatial movement action, S_{EP} is reset to "nothing recalled."

The agent has two actions available to it with which it can interact with its episodic memory. The first action, "cue retrieval," finds the most recent instance of the agent's current state in the list and sets S_{EP} to that state (or to the "nothing recalled" state if the current state is not found in the list). The other action, "advance retrieval," sets S_{EP} to the state following its current state in the list (or, if S_{EP} is "nothing recalled," it remains unchanged).

Simulations including the episodic memory system also include a WM system to buffer episodic retrieval. Effectively, this introduces a third action that stores the current episodic memory state S_{EP} in WM.

For some of the simulations below, S_{EP} forms memories of S_L to allow retrieval and replay of sequences of visited locations. The exceptions are in the tone-cued alternation task, S_{EP} forms memories of both location and auditory input, in the nonmatch to position and sample tasks, S_{EP} forms locations of both location and visual input regarding barriers and levers, respectively, and in the odor sequence disambiguation task, S_{EP} forms memories only of odors.

The function of these two actions and their interaction with the other environments in the simulations are summarized in Table 2.

This mechanism maps closely onto the path readout model of hippocampus proposed by Hasselmo and Eichenbaum (2005), but differs in that this earlier model treated cuing retrieval and advancing multiple steps in memory as a single process that occurred automatically at every step of a simulation, instead of treating the two as separate actions that must be actively selected, as in the present framework. Thus, the important difference is that retrieval of episodes is now under the control of the agent instead of being automatically performed on every step. Although, here, as in that model, encoding still happens automatically and constantly (see also Morris and Frey, 1997).

TABLE 2.

Summary of States and Actions in Episodic Memory Agent Simulations

States	Actions	State transition
S_L, a sensory state	Motor action (e.g., N/S/E/W movement, lever press, and dig for reward in cup of sand)	$S_L \leftarrow$ sensory state S_{WM} unchanged $S\text{-}_{\mathrm{EP}}$ cleared
S_{WM}, a sensory state or empty	Add to buffer	S_L unchanged $S_{\mathrm{WM}} \leftarrow S_{\mathrm{EP}}$ $S\text{-}_{\mathrm{EP}}$ unchanged
S_{EP}, a sensory state or empty	Cue retrieval	S_L unchanged S_{WM} unchanged $S_{\mathrm{EP}} \leftarrow S_L$
	Advance retrieval	S_L unchanged S_{WM} unchanged $S_{\mathrm{EP}} \leftarrow$ next state in temporal sequence

The content addressability in this context also differs somewhat from the traditional definition. Content addressability as described in Marr (1971) is what has also been described as pattern completion (O'Reilly and McClelland, 1994), a process by which a subset of the representation of some cue or memory elicits the entire memory directly. The present mechanism introduces an intermediate indexing step, which still allows for a subset of a memory to elicit the complete memory via the index, but also provides an ordering on the indices so that subsequent memories can be read out in the proper order. The intermediate step has some similarity with the context signal used in some models of verbal WM (Burgess and Hitch, 2005), which are more similar to our episodic mechanism than to our WM mechanism.

Rewards and Limitations on Actions

The agent was free to take any action in any state; however, some actions did not change the agent's state and result in a reward of −1. These included attempting to move out of the environment or into a barrier, to store something in WM that is already stored, and to advance episodic retrieval without first cuing retrieval. Also, as is common in behavioral tasks using T-mazes, the agent was prevented from moving backward, here by giving a reward of −1 if the agent attempted to re-enter a location it had just left. In general, attempting to take any "illegal" action resulted in this negative reward and no state change. Finally, four tasks (all except spatial alternation and tone-cued spatial alternation) had an upper limit on the number of steps allowed, 100. When this limit was reached, the episode ended and was counted as incorrect. For most tasks, the agent received a reward of +9.5 for a correct response, −6 for an incorrect response, and −0.05 for all other actions to encourage the agent to find the shortest solution to a task (particularly important when memory actions are available, to avoid the agent repeatedly taking those actions instead of taking spatial movement actions). For the odor sequence disambiguation task, the agent received a reward of +1 for reaching a correct goal state, −5 for an incorrect response, and −0.5 for all other actions. These values were selected to produce learning at a reasonably fast rate (the odor sequence task required different values). In many cases, faster learning could be achieved by setting task-unique reward values.

RESULTS

Before we present results of our simulations of various tasks, one aspect of episodic memory should be emphasized to make the performance figures more clear. All simulations with episodic memory also included a WM system to allow the agent to hold the retrieved episodic memory online to guide behavior (a requirement for proper learning in these simulations, this is similar to the context-reinstantiation idea described in Redish, 2001). In general, if an agent in a nondelayed task with WM could solve a task, so could an agent in a simulation with episodic memory (an agent with episodic memory may, for example, react differently at a novel state or a state not visited for a large number of steps, as these cases may produce "nothing recalled" states in response to the cue retrieval action). It was in delayed tasks, wherein WM was cleared at locations as described below (shown as striped squares in the figures), that the difference between agents with episodic memory and WM became clear. If an agent with episodic memory could solve a delayed task, then it must be that the agent was using episodic retrieval as a means of retrieving an old memory sequence, instead of simply using its WM component to hold a state online. Table 1 summarizes the states and actions available to a WM agent; Table 2 summarizes those for an episodic memory agent.

For all tasks, there were ultimately two possible responses of which one was correct, thus chance performance was at 50% correct. In some cases, as described below, an agent consistently performed well below chance and in one case the agent was above chance but not at 100% performance.

Spatial Sequence Disambiguation

We first consider a task that is a simple demonstration of the use of WM. Spatial sequence disambiguation was implemented here in an I-shaped maze, shown in Figure 1, top, comprising a vertical stem with left and right horizontal arms on both ends. The agent was randomly started in one of the two bottom arms and forced to enter the vertical segment by a barrier blocking the other start arm. It received a positive reward only in the opposite side arm on the top of the maze. Whether the agent was rewarded or not, it was removed from the maze and repositioned on a random start arm for the next trial. We simulated two versions of this task. In the delayed version, the agent's WM was cleared when it reached the bottom of the stem to represent a delay period imposed by an experimenter.

Figure 2 demonstrates the way the agent with WM could solve this task by showing different sets of action values, depending on which location was held in WM, represented in the figure by a square of small circles around the location being held. In particular, for each location on the starting arms in memory, the choice point had one action with a positive value (dark arrows), indicating the direction the agent should go to receive a reward. It should be emphasized that these action values are learned and that the action values learned for each value

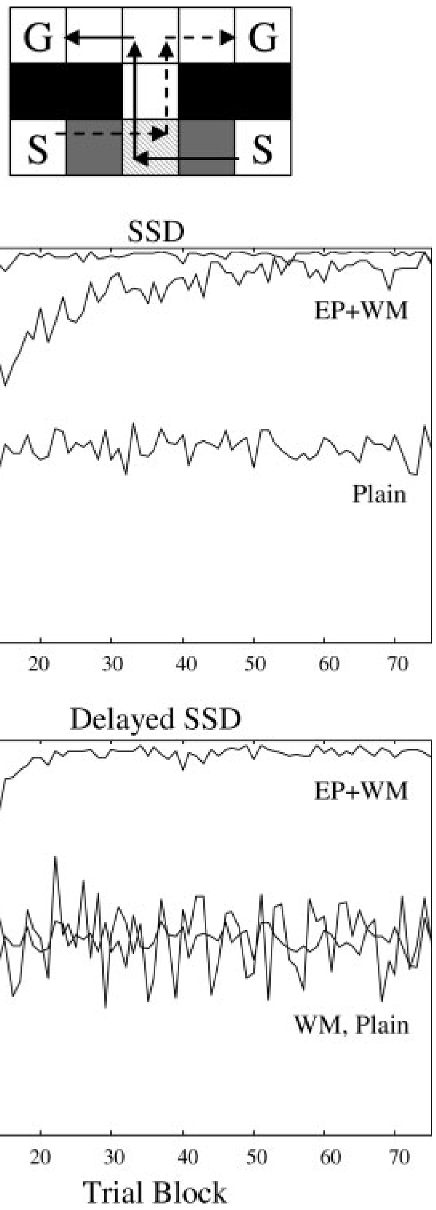

FIGURE 1. Spatial sequence disambiguation (SSD). Top, environment used in this task. *S* indicates starting position, and *G* indicates a position where a reward may be available. Filled black squares are static barriers. Filled gray squares are movable barriers. The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. In the nondelayed task (middle), the agents with working memory learn slightly faster than the agents with episodic memory on average. In the delayed task (bottom), the working memory of the two agents with memory systems is cleared before each episode, reducing interference across trials. Performance in each block of 20 trials is the number of correct responses divided by the total number of responses.

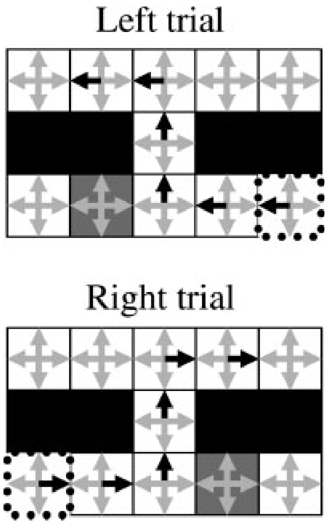

FIGURE 2. **Working memory's effect on policy. The arrows show the agent's learned action values corresponding to the two trial types ("should go left" trial on top, "should go right" trial on bottom). The state in the agent's working memory is indicated by a square of circles, which determines the indicated action values. Black arrows represent positive action values, light gray arrows represent negative or zero action values. Filled black squares represent impassable barriers; filled gray squares represent movable barriers.**

of WM contents are independent. Figure 1 shows the performance on the delayed and nondelayed versions of this task. For each version of the task, we show performance for an agent with episodic memory and WM (EP + WM), an agent with only WM, and a normal agent with no extra memory systems (plain). As shown in the middle in that figure, both the episodic and WM agents were able to solve the nondelayed task, whereas only the episodic agent could solve the delayed version (Fig. 1, middle and bottom). The standard agent could not learn the task in either version. As mentioned earlier, because the episodic agent could solve the delayed version, it must have been using something more complex than a straightforward WM approach. Because the trial types (on which arm the agent starts, and therefore on which arm it is rewarded) were randomly selected on each trial, it may be surprising that episodic memory could solve this task. At the choice point, the agent's WM had been cleared, and so it must have relied on memory of a state that followed its last visit to its current state. However, the last visit to the choice point had been on the previous trial, so the reward arm it selected on that trial was independent of the correct reward arm for the current trial. The strategy that the agent discovered was, in fact, to retrieve the memory of that previous trial and then to advance retrieval multiple steps, replaying the episode up to and past the rewarded state at the end of the arm, which then brought its memory to the beginning of its current trial. It was thus able to recover at which location it began the trial and could make its decision based on that memory.

This strategy worked in the simulations because there was effectively no intertrial interval: each trial began immediately after the previous one ended. Because experiments with real animals often use a delay between trials, either so that the experimenter may reset the apparatus for the next trial or to reduce interference across trials, real animals may not be able to use this strategy.

Spatial Alternation

The next task we consider is spatial alternation (Wood et al., 2000; Lee et al., 2006; Ainge et al., 2007). This task took place in the environment shown in Figure 3, top. The agent began at the bottom of the center hallway in the maze and proceeded up to the choice point. On the first trial, the agent could turn either left or right and received a positive reward at the end of the arm. The agent then continued along a return arm and back to the starting point and then again proceeded up the center hallway to the choice point (at the starting point, the agent was prevented from entering the side hallway instead of the center hallway by use of a barrier). After the first trial, the agent received a positive reward if it chose the opposite direction as on the previous trial, otherwise it received a negative reward. The delayed version of this task was identical, except that the agent's WM was cleared on every visit to the start position (bottom of the stem). Performance in this task is plotted in Figure 3.

As in the previous task, this task is non-Markov because the correct turn at the choice point depends on the agent's previous response. However, with a WM action available, the agent could buffer a location on the return arm while heading back to the start position and maintain it in memory until the choice point was reached, thus allowing disambiguation as to the trial type. In this way, WM can also be thought of as a sort of memory for ongoing context. The agent learned different policies for each item that may be stored in WM, which may be interpreted as learning behaviors appropriate for a given context. Specifically, this works because, due to the factored state representation, the agent had many different policies depending on which location state was in its WM. Alternatively, episodic memory could be used to solve the task by cuing retrieval at the choice point, advancing one step in memory to retrieve the previous response made, then making the opposite response (sometimes the simulated agents would instead, for example, cue retrieval one step before the choice point, then advance retrieval by two steps to recall the previous response). In the delayed version, only an agent with episodic memory could solve this task, as shown in Figure 3, right, in agreement with experimental work using hippocampal lesions (Ainge et al., 2007).

Electrophysiological recordings of neurons in animals performing this and similar tasks have shown that neurons in the hippocampus sometimes discriminate between left-response trials and right-response trials (Wood et al., 2000; Frank et al., 2000; Ainge et al., 2007), but in other cases the hippocampal cells do not discriminate (Lenck-Santini et al., 2001; see also Bower et al., 2005). The fact that the spatial alternation task could be successfully performed with either episodic or WM may provide insight into the differences seen in neuronal responses. This suggests that subtle differences in training or task demand in those studies may bias the animal to use a WM strategy over an episodic memory strategy or vice versa.

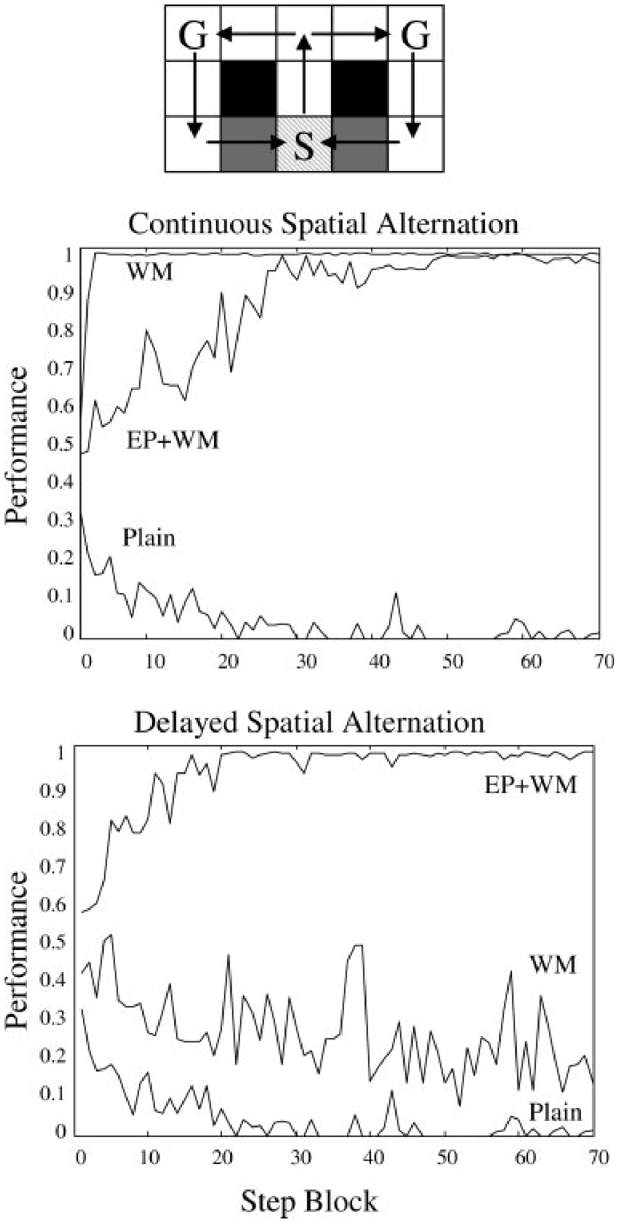

FIGURE 3. Spatial alternation. Top, environment used in this task. *S* indicates starting position and *G* indicates a position where a reward may be available. Filled black squares are static barriers. Filled gray squares are movable barriers. The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. Performance in each block of 2,000 steps is the number of correct responses divided by the total number of responses.

Nonmatch to Position

Our simulated NMP task took place on a T-maze as in the spatial alternation task, but without the return arms leading from the reward states to the starting position (Griffin et al., 2007), as shown in Figure 4, top. Each trial in this task consisted of two stages, both of which began with the agent being placed

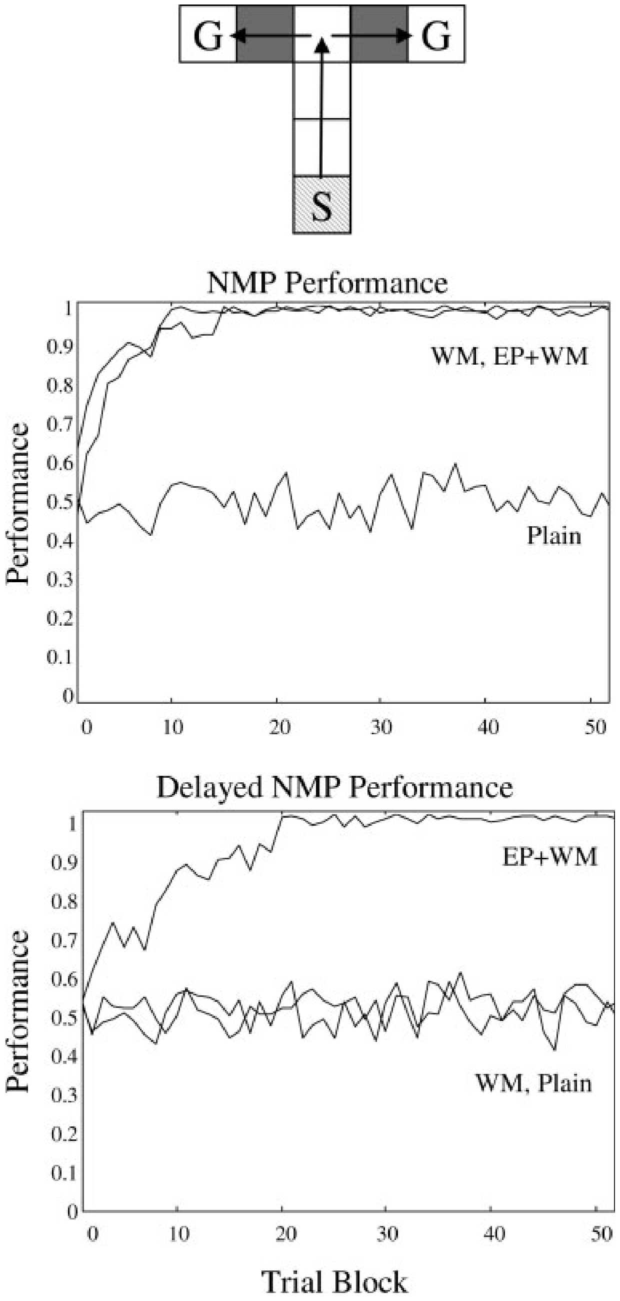

FIGURE 4. Nonmatch to position (NMP). Top, environment used in this task. *S* indicates starting position and *G* indicates a position where a reward may be available. Filled gray squares are movable barriers. The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. The agent with episodic memory is always able to perform the task, and the agent with no memory is always at chance, while the working memory agent can perform correctly only in the nondelayed version (middle). Performance in each block of 40 trials is the number of correct test stage responses divided by the total number of test stage responses.

at the bottom of the stem and proceeding to the choice point. In the sample stage, one of the two arms of the maze at the choice point was blocked, forcing the agent to go in a particular direction. The agent received a reward at the end of the arm it was forced to enter, then was placed back at the starting point to begin the choice stage. In the choice stage, neither arm was blocked at the choice point. If the agent entered the same arm, it had entered on the sample stage (a match response), it received a negative reward at the end of the arm; if the agent entered the opposite arm (a nonmatch response), it received a positive reward at the arm's end. The blocked arm on the sample stage was randomly selected on each trial. In the nondelayed version of this task, the agent's WM was cleared before it was placed at the start position in each sample stage but not each choice stage (this improved the learning rate by eliminating interference across trials). For the delayed version, the agent's WM was cleared before both sample and choice stages. The choice point and the state immediately below it were each represented by one of three different states: choice point with right arm blocked, choice point with left arm blocked, and choice point with no arms blocked. These correspond to a conjunctive state of spatial location and barrier position to reflect differences in visual or tactile input a real animal would receive in such a task.

As shown in Figure 4, middle and bottom, this task could be solved by episodic memory in both the delayed and nondelayed versions and by WM in the nondelayed version. Using WM, the agent could buffer a location on the reward arm during the sample stage, which could then indicate the correct response at the choice point in the choice stage, allowing success only in the nondelayed version (Fig. 4, middle).

If the agent did not receive state information reflecting the presence of barriers, which signal the sample stage, performance was significantly impaired (unpublished observations). This requirement of representing task stage is supported by the hippocampal recordings of rats performing a delayed NMP task on a T-maze, where many units show modulation by task stage (Griffin et al., 2007). Our simulated task differs slightly from that of Griffin et al. (2007) in that we began each stage of the task with the agent at the bottom of the stem, whereas rats in the Griffin study began the sample stage on a pedestal with a ramp leading to the stem and they began the choice stage at the bottom of the stem. As in our spatial sequence disambiguation task, the different starting positions could be held in WM to differentiate the two task stages, improving performance and suggesting a means by which the stage selective firing in real rats could have come about.

Nonmatch to Sample

We also simulated a nonmatch to sample task in which responses were made on two levers in an operant chamber (Hampson and Deadwyler, 1996). Here, each trial consisted of a sample stage in which only one of the two levers was available for making a response, then a choice stage in which both levers were available. The agent received a negative reward for making a response at the same lever as had been available during the sample stage and a positive reward for making a response at the other lever. As shown in Figure 5, top, the two levers were positioned at opposite corners of the environment along the same wall. After making a response at a lever, the animal was moved back to the starting position (a slight difference from experi-

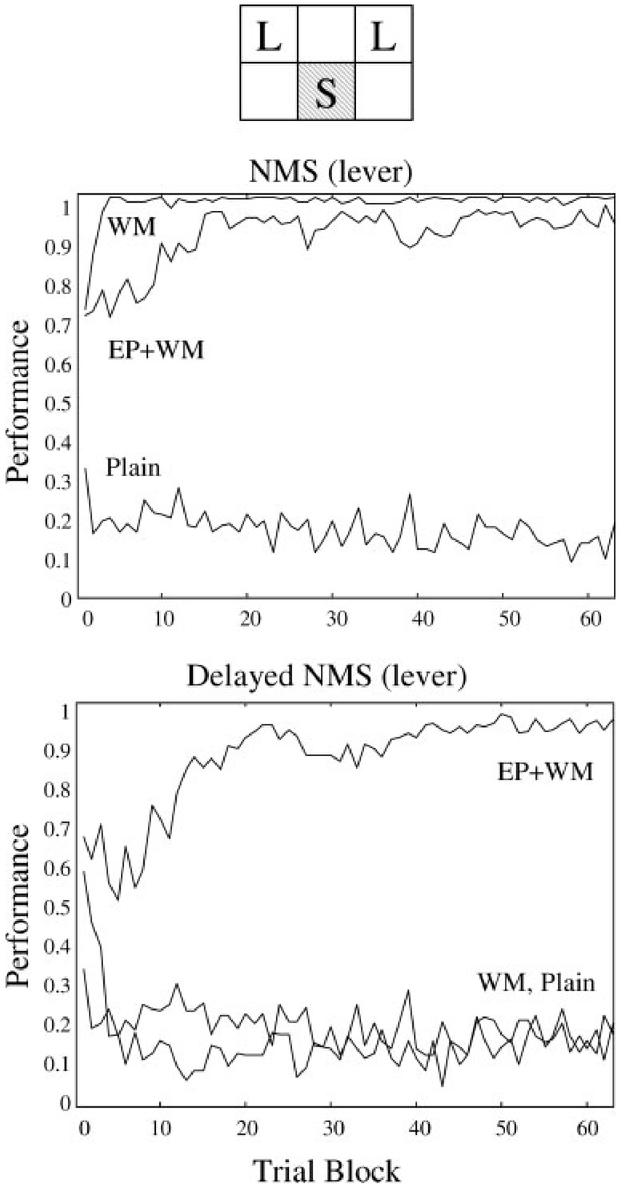

FIGURE 5. Nonmatch to sample (NMS) with levers as stimuli. Top, environment used in this task. *S* indicates starting position. *L* indicates a response lever. The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. The agent with episodic memory is always able to perform the task at greater than 80%, and the agent with no memory is always at chance, while the working memory agent can perform correctly only in the nondelayed version (middle). Performance in each block of 20 trials is the number of correct test stage responses divided by the total number of test stage responses.

mental versions of this task which may require the animal to, e.g., make a nose poke at a sensor at a location distant from the pair of levers). In the delayed version of this task, the agent's WM was cleared before starting the choice stage (WM was cleared before the sample stage in both task versions to eliminate interference between trials). To simulate visual input regarding the state of the levers, the agent had separate state representations for the sample stage with the left lever extended, the sample stage with the right lever extended, and the choice stage with both levers extended. The spatial environment comprised two rows and three columns of locations, two of which are locations at which a lever may be present. Thus, there are 18 location states in total (six locations time three different task stage conditions). The need for separate states for sample and choice stages was mentioned earlier, in the NMP task section, and will be discussed again later in the article.

In principle, both versions of this task were solvable using episodic memory; however, the nondelayed version was more readily solved using WM, as shown in Figure 5, middle. By maintaining in WM the lever at which a response was made during the sample stage, the agent could learn to proceed to and respond to the opposite lever during the choice stage. When the agent could not solve the task, its performance was well below chance. This occurred because the agent rarely made responses, and the episode eventually was terminated when the upper time step limit was reached.

Performance did not depend on the nonmatch rule used here and in the NMP task above; both tasks were solvable with either a match or a nonmatch rule (unpublished observations).

Odor Sequence Disambiguation

Our odor sequence disambiguation task was modeled after the task from Agster et al. (2002), though some simplifications were made. As shown in Figure 6, top, the simulated task took place on a linear track with five pairs of odors spaced along its length. On each trial, the agent was to make a response to one odor of each pair. On sequence 1, the agent was rewarded for responding at each of the odors 1–5 (the "top" odors). For sequence 2, the agent was rewarded for selecting each of the odors 6, 7, 3, 4, and 10. Thus, the two sequences overlap in that odors 3 and 4 were always correct, and odors 8 and 9 were never correct. After selecting the correct odor from a pair, the barrier in front of the rat was removed and it could advance to the next pair. Except for the final pair, if the agent attempted to dig at the incorrect odor, it was allowed to correct itself (in fact, it had to before it could continue). On the final pair, the agent could dig at either odor and the trial was recorded as correct or incorrect, depending on the response and the sequence type. In the delayed version of this task, the agent's WM was cleared before the final pair was presented.

The agent had five actions available: move forward, sample top odor, sample bottom odor, dig at top odor, and dig at bottom odor. Each odor had three corresponding states: sampling odor, successfully digging at an odor (if it selected the correct odor for a pair), and unsuccessfully digging at an odor (if it

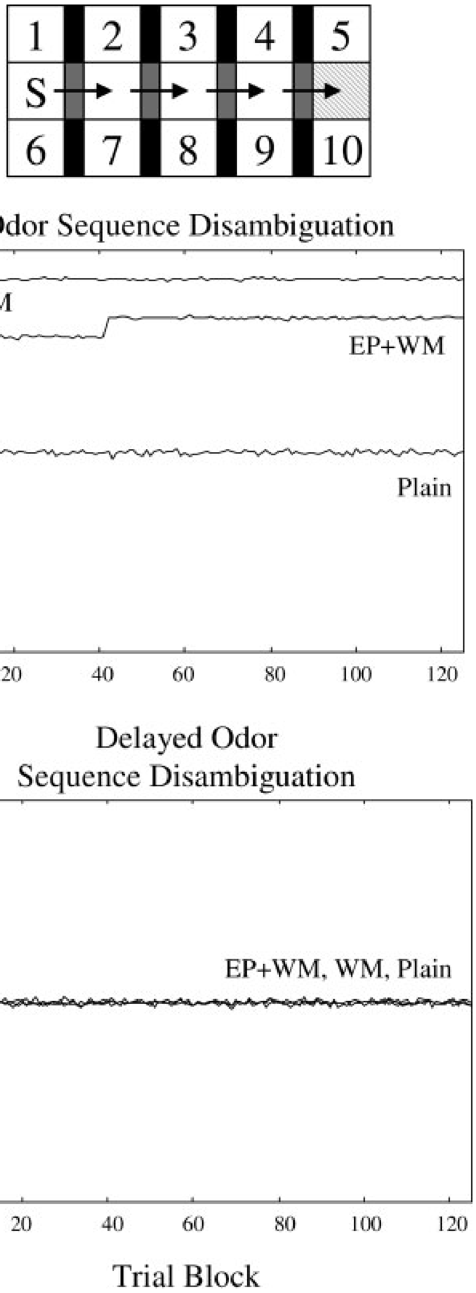

FIGURE 6. Odor sequence disambiguation. Top, environment used in this task. *S* indicates starting position. Each number indicates a different odor and the thin rectangles represent barriers that were not explicitly included as locations (black are static barriers and gray are movable barriers). The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. In the delayed task (bottom), the working memory of the two agents with memory systems is cleared before each episode, reducing interference across trials. Performance in each block of 1,000 trials is the number of correct responses divided by the total number of responses.

selected the incorrect odor for a pair). Until the agent sampled an odor from the current pair, it was in a state simply corresponding to its location. After sampling an odor, the agent remained in the state of smelling that odor until it sampled the other or advanced to the next odor pair after successfully digging. In some experimental versions of this task, the scented cups had perforated lids that the rat had to remove before it may dig. To simulate these lids, we used the restriction that until the agent had sampled an odor, it could not attempt to dig at it (thus it was able to smell the odor before it could dig in the cup). Without this restriction, the agent was still able to learn the task, but at a slower rate (unpublished observations).

As with the spatial sequence disambiguation task, this task required the agent have a form of WM to perform correctly. Specifically, the agent had to hold in WM the state of having successfully dug at one of the two initial odors to make the correct choice on the final pair. For this reason, an agent with only WM could not solve the delayed version of this task, shown in Figure 6, bottom.

The agent with episodic memory had two possible strategies to use, as in the spatial sequence disambiguation task. In the nondelayed version, it could simply take a WM approach to perform successfully (Fig. 6, middle). An alternative approach might be for the agent to use episodic retrieval with one of the cups in the final pair as the cue, which could retrieve a memory of the end of the previous trial. Then the agent could advance retrieval by multiple steps until it reaches a memory of successfully digging in one of the cups in the first pair on the current trial. Real animals may not be able to use this strategy for the same reason given in the sequence disambiguation task: in the simulations there were no intervening stimuli between the end of one trial and the beginning of the next, whereas behavioral experiments in laboratory animals often did contain such stimuli because of the presence of an intertrial delay. Agents required a significant amount of training to learn this strategy, which further argues against the likelihood of animals using this approach (so the simulation in Figure 6 was not run long enough to show the agents with episodic memory learning to correctly perform the task). In this case, the first state the agent experiences on a trial did not disambiguate the trial type, so that it was forced to continue advancing retrieval until it reached a state where an odor was successfully dug. The increased number of ambiguous states leading up to the successful digging made the agent's performance much more sensitive to random exploration either during retrieval at the end of the trial or during behavior at the beginning of the trial itself.

Tone-Cued Spatial Alternation

Next, we consider a tone-cued spatial alternation task. To our knowledge, this task has not been used in the literature, so that the present simulation results constitute a prediction of this model. This task took place on a figure-8 maze as in the earlier spatial alternation task, shown in Figure 7, top. The agent began at the bottom of the center hallway in the maze and proceeded up to the choice point (the agent was prevented

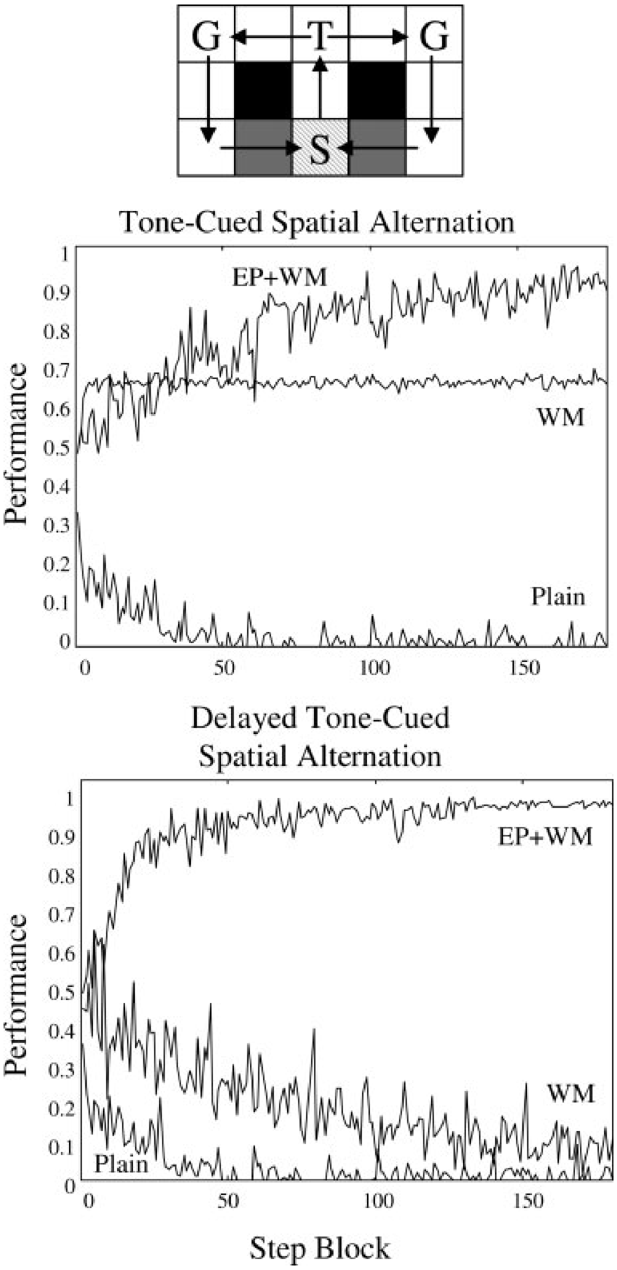

FIGURE 7. Tone-cued spatial alternation. Top, environment used in this task. *S* indicates starting position, and *G* indicates a position where a reward may be available. *T* indicates the location where one of two tone-cues is played. Filled black squares are static barriers. Filled gray squares are movable barriers. The striped square is where working memory is cleared as described in the text. Middle and bottom, average performance of 10 runs with both episodic memory and working memory (EP + WM), working memory only (WM) and an agent with no memory systems (plain) in the nondelayed (middle) and delayed (bottom) versions of the task. In the delayed task (bottom), the working memory of the two agents with memory systems is cleared before each episode. Performance in each block of 3,000 steps is the number of correct responses divided by the total number of responses.

from directly entering a return arm from the start point with the use of two barriers, as shown in the figure). Here, one of two tones was randomly sounded (implemented in the simulations as two different states due to the factored state representation: choice point with tone 1 and choice point with tone 2). The agent could go in either direction on its first exposure to each tone and receive a positive reward at the end of the arm. The agent then followed the return arm back to the bottom of the center hallway and proceeded to the choice point, and a random tone was again played. Now the agent was only rewarded for choosing to move in a direction opposite to its response when the current tone was last presented. That is, if the agent last made a left response when tone 1 was played, the next time tone 1 is played the agent had to make a right response, regardless of how it had responded at any number of intermediate presentations of tone 2. Likewise, each time tone 2 was presented, the agent had to make the opposite response as on the previous presentation. In the delayed version of this task, the agent's WM was cleared on each visit to the bottom of the stem. The use of two tones as the cue was selected arbitrarily. If this experiment were attempted with animals, using cues from different modalities (such as a tone and a light) may reduce interference and make the task easier to learn.

To perform the task correctly, the agent was forced to use its episodic memory actions to retrieve the last presentation of the current tone and advance one step in memory to discover which direction it previously went, then choose the opposite direction. WM could not correctly solve this task as the correct choice does not depend on the immediately preceding trial, but rather on the most recent trial in which a given tone was presented.

As indicated in Figure 7, middle, however, in the nondelayed task, the WM agent was above the chance level and the agent with no memory systems was well below chance. The plain agent tended to choose the same arm over and over again, thus producing a series of incorrect trials, until it selected a random action at the choice point or the action values became low enough that the agent switched responses, receiving two correct trials followed by another series of incorrect trials.

With WM, though, a better strategy was available: simple spatial alternation. Consider the following: the agent is at the start position, having just returned from making a left response, for example. There are three possible situations the agent might face. It could be that, for both tones, the agent was to go right on the previous trial, so regardless of which tone was played, the agent just made an error (thus, in a sense, the rewards of both tones are currently on the right reward arm). Or it could be that the reward for tone 1 is on the left reward arm, and the reward for tone 2 is on the right reward arm (in which case it must be that tone 2 was played on the previous trial, because if tone 1 had been played the agent's response would have been correct, so tone 1's reward would now be on the right arm). Similarly, it could be that the reward for tone 2 is on the left reward arm and tone 1 on the right arm (the previous case, with tones 1 and 2 reversed). However, it could not be the case that both tone 1 and tone 2 currently have their rewarded arm on the left, because the agent just came from the left arm, so that one of the two would have been switched to the right arm.

With an alternation strategy, its next response will be to choose the right arm, regardless of which tone is played. In the first case, for either tone 1 or tone 2, the agent will be rewarded on the right arm, so one-third of the time it is guaranteed to receive reward. In the other two cases, its probability of receiving a reward on the right arm is 0.5, since that is the probability that the tone corresponding to the reward on the right arm will be played. Thus there are six possible outcomes of which four are favorable, so the agent will receive a reward on 2/3 of the trials, as shown in Figure 7, middle, which is above chance level. In contrast to the partial reward strategy for agents with only WM, the agent with episodic memory received a higher average reward than the WM agent on both the nondelayed and the delayed version of the task.

DISCUSSION

The goals of this work were twofold. First, we sought to extend the reinforcement learning framework to include memory systems analogous to those thought to exist in animals, so that common tasks used in behavioral experiments could be simulated in a single, quantitative framework. Second, we sought to determine the way that slight differences in these tasks can change the demands on different subsystems in the brains of animals so as to inform future experimental work. To the extent that our abstractions of these memory systems are correct, we have shown that the information required to make correct choices may require different systems whose utility may overlap, but which can be dissociated by specifically designing tasks to preclude the use of all systems except those of interest. We now discuss these points in more detail.

Working Memory Versus Episodic Memory

A formal description of working and episodic memory may guide future research on memory function by providing a means to evaluate potential tasks to see which forms of memory can be successfully used in performing them.

If a task can be solved based on which of a set of states was most recently experienced, then WM can solve the task (though the time required to learn the task increases as the amount of time or number of states between the choice point and the most recently experienced disambiguating state increases). If a task can be solved based on the states that previously followed some given experienced state on the current trial, then episodic memory can solve the task (though the time required to learn the task increases as the number of retrieval steps required to access the disambiguating state increases).

As was demonstrated, for instance, by the spatial sequence disambiguation task, even though a task may appear to be solvable with only one memory system, there may be unexpected (sometimes artifactual) ways that other memory systems can be used instead. Even with the simple systems presented here this was possible, so with more complex and realistic memory systems, the possible strategies may greatly increase in number.

Multistage Matching Tasks

A common feature of tasks used in behavioral or electrophysiological studies is the presence of multiple, distinct stages. For example, a nonmatch-to-sample (NMS) task consists of a sample stage during which a stimulus is presented, then a choice stage in which multiple stimuli are presented, either simultaneously or consecutively, during which the animal must respond to the nonmatching stimulus (i.e., the stimulus not presented during the sample stage). A problem in implementing such tasks in the reinforcement learning framework is that the animal must know which stage it is in to make the proper response. In an NMS task, the agent must make a response to the sample stimulus in the sample stage, but not to that stimulus in the choice stage, so without task stage information as part of its state, it cannot learn such a task. When real animals perform such tasks, there is often stage information provided to them in some form (i.e., in primate matching tasks the animal often begins the trial by pulling on a lever which it releases to respond during the choice stage (Miller et al., 1996), and in rodent NMS tasks with levers (Hampson and Deadwyler, 1996), there is one lever available during the sample stage and both available during the choice stage-in both cases sensory information or memory of recent actions can signal task stage). In such cases, one can simply include an extra element in an agent's factored state representation that indicates task stage, but in other tasks where the animal must learn on its own which stage it is in, the proper reinforcement learning approach is not clear. Indeed, in some tasks, each trial is both a sample and choice stage, such as the continuous odor nonmatching task used by Wood et al. (1999) and Young et al. (1997) or the analogous *n*-back task used in humans (Stern et al., 2001).

In this work, we have simulated only two matching tasks: a delayed NMP task in a T-shaped maze and a delayed non-match to sample task using levers in an operant chamber. There are, however, many other variants of matching tasks, each of which has its own idiosyncrasies when considered in the reinforcement learning framework. In primate experiments, the stimuli in the matching tasks tend to be either visual stimuli on a computer screen (Riches et al., 1991; Miller et al., 1996) or objects at spatial locations in front of the animal (Buckmaster et al., 2004). In experiments with rodents, the stimuli are often levers (Hampson and Deadwyler, 1996), cups of scented sand (Dudchenko et al., 2000), objects at spatial locations (Kirwan et al., 2005), or spatial locations themselves (Zironi et al., 2001).

One dimension along which we can categorize matching tasks is whether the stimuli are unique on every trial or are repeated over the course of the experiment. In tasks with levers it is clear that the levers' identities are always the same on every trial, and, with odors, it is usually the case that the odors are eventually repeated (as the experiment would otherwise require a very large library of odors to be used). When objects are used, either as visual images on a computer screen or real objects placed around the animal, the objects are sometimes reused (Rapp and Amaral, 1989) and sometimes always unique (Buckmaster et al., 2004). This distinction is relevant to simulations using the reinforcement learning framework, as such an agent makes its decisions by selecting the action with the highest value for its current state. If the stimuli are unique on each trial, then on each trial the animal will not have learned any actions to take from the state corresponding to a specific object, and it will never fully learn the task. Suppose that the agent learns to hold the unique sample stimulus in WM (perhaps by learning the utility of ignoring the object's identity and simply taking a "set WM" action in the sample stage), the agent must still be able to detect a match or nonmatch during the choice stage of the task. Even if, on the first trial of a match task, the agent makes a correct response to the factored state of (A in WM, A being experienced), on the next trial the agent will have no action values for the factored state of (B in WM, B being experienced). Thus, in addition to a WM system, an agent needs a mechanism by which it can detect a match between its WM and its sensory input. One potential means of solving this problem is a role-filler mechanism as is used in analogical reasoning models (Hummel and Holyoak, 1997). It should be noted that although increased or decreased neuronal firing rate to matching stimuli has been reported in temporal cortex (e.g., Miller et al., 1993; Holscher et al., 2003), this neuronal activity does not solve the problem noted earlier, as the increased firing is stimulus dependent and so does not provide the abstract signal of "matching" needed for the present purposes. These firing rate differences, however, are thought to reflect familiarity and so a mechanism by which they can be used to influence decision making will likely be very important to modeling this type of task.

Another dimension of matching tasks is the question of whether the identities of the choice stimuli are made known simultaneously or consecutively. For example, in the T-maze delayed NMP task simulated earlier, when the agent is at the choice point in the maze, both of the possible responses are available to it at the same time (go left or go right) and are consistent across trials. Making a response then is as simple as selecting the action with the highest action value at that state (where, through some memory mechanism, the state should be different on right trials vs. left trials). The delayed match to sample task also described earlier is somewhat in between the simultaneous and consecutive extremes. The identities of the stimuli are consistent across all trials (i.e., the left lever is always the left response) but there is no single location from which both the left and right lever responses are immediately available. Still, as the agent starts the choice stage knowing which response it should make, it can take the appropriate series of actions to bring it to the correct lever. The distinction can be made more clear by contrasting it with a purely consecutive task, such as an odor nonmatch task. Suppose the agent is presented with a cup of scented sand in the sample stage, then is presented with two scented cups of sand (one with the same odor from the sample stage and one with a different odor as well as a food reward) for the choice stage in randomized spatial positions (Dudchenko et al., 2000; McGaughy et al., 2005). In this case, the agent from afar is not aware of which cup has which odor, and thus which cup contains the food reward. The agent must select one cup to approach, then only after sampling the odor may the agent make a response (if at the nonmatching odor) or withhold a response and approach the other cup (if at the matching odor). From a reinforcement learning standpoint, this is a relatively complex task. It appears to require first that the agent have actions to select one of the two cups of sand as a destination which then alter the action values of the location states to direct it to the cup. The agent must then learn to select the other cup as a destination if the odor in memory matches the odor at that cup or to make a response if the odors do not match. This requirement is not trivial, as if there are more than a few odors used in the task, the number of conjunctive states for which the odor in memory does not match the experienced odor can be quite large, and, for each such state, the agent must learn to make a response. Here, a utile distinction learning approach (McCallum, 1995) may be helpful, allowing the agent to distinguish between the two states, one being the triple (choice stage, sampling odor, match detected) and the other being the pair (choice stage, sampling odor), with the former taking precedence over the latter.

Relationship to Other Models

To our knowledge, this is the first report showing that retrieval of episodic memories during behavior increases the domain of problems reinforcement learning systems can solve, but it is not the first use of hippocampal mechanisms in reinforcement learning. Foster et al. (2000) treated hippocampal place cells as radial basis function units for representing space, associating action values to individual place cells and linearly summing them, scaled by the activation of each unit. Arleo and Gerstner (2000) used a similar approach, using a form of Q-learning (Watkins, 1989) to modify the strength of synapses from hippocampal place cells representing location to neurons in the nucleus accumbens representing actions.

Johnson and Redish (2005) also used place cells as radial basis functions, but showed further that offline (during awake states, but not while engaged in the task) replay of activation in hippocampus can be interpreted as a form of the Dyna-Q reinforcement learning algorithm (Sutton, 1990), which allows the agent to learn tasks more quickly. Recent reports of hippocampal replay when rats are idle during a task (Foster and Wilson, 2006; Jackson et al., 2006) also support this suggested role.

These two functions are not at all exclusive of the present suggestion of prefrontal-initiated hippocampal retrieval of episodes for guiding behavior. In fact, all three proposals could easily work together to combine all their individual advantages. The fact that these aspects of hippocampal function work well within the reinforcement learning framework, along with the suggestions that the basal ganglia may be implemented with an reinforcement learning-like mechanism suggest that this may be a fruitful theoretical approach for future research.

Other modeling work (Gaussier et al., 2002; Banquet et al., 2005) has interpreted hippocampal place cells as representing transitions between states, using these to form graphs to represent trajectories through space for navigation. These graphs are similar to those used by Hasselmo (2005) and Zilli and Hasselmo (2005) in neural implementations of reinforcement learning-like algorithms for navigation and behavior. The reinforcement learning framework provides an abstract framework for action selection through which different neural implementations can be tested and which can guide research toward the correct neural implementation.

Acknowledgments

The authors thank Lisa Giocomo and Amy Griffin for helpful comments on an early draft of this manuscript, and two anonymous reviewers for their thorough and tireless efforts to make this manuscript as good as possible.

REFERENCES

Agster KL, Fortin NJ, Eichenbaum H. 2002. The hippocampus and disambiguation of overlapping sequences. J Neurosci 22:5760–5768.

Ainge JA, van der Meer MA, Langston RF, Wood ER. 2007. Exploring the role of context-dependent hippocampal activity in spatial alternation behavior. Hippocampus 17:988–1002.

Arleo A, Gerstner W. 2000. Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity. Biol Cybern 83:287–299.

Baddeley AD. 1986. Working Memory. Oxford: Clarendon Press.

Baddeley AD, Hitch G. 1974. Working memory. In Bower GH, editor. The Psychology of Learning and Motivation: Advances in Research and Theory. New York: Academic Press. pp 47–89.

Banquet JP, Gaussier P, Quoy M, Revel A, Burnod Y. 2005. A hierarchy of associations in hippocampo-cortical systems: Cognitive maps and navigation strategies. Neural Comput 17:1339–1384.

Bower MR, Euston DR, McNaughton BL. 2005. Sequential-context-dependent hippocampal activity is not necessary to learn sequences with repeated elements. J Neurosci 25:1313–1323.

Buckmaster CA, Eichenbaum H, Amaral DG, Suzuki WA, Rapp PR. 2004. Entorhinal cortex lesions disrupt the relational organization of memory in monkeys. J Neurosci 24:9811–9825.

Burgess N, Hitch G. 2005. Computational models of working memory: Putting long-term memory into context. Trends Cogn Sci 9:535–541.

Dudchenko PA, Wood ER, Eichenbaum H. 2000. Neurotoxic hippocampal lesions have no effect on odor span and little effect on odor recognition memory but produce significant impairments on spatial span, recognition, and alternation. J Neurosci 20:2964–2977.

Foster DJ, Wilson MA. 2006. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. Nature 440:680–683.

Foster DJ, Morris RGM, Dayan P. 2000. Models of hippocampally dependent navigation using the temporal difference learning rule. Hippocampus 10:1–16.

Frank LM, Brown EN, Wilson M. 2000. Trajectory encoding in the hippocampus and entorhinal cortex. Neuron 27:169–178.

Gaussier P, Revel A, Banquet JP, Babeau V. 2002. From view cells and place cells to cognitive map learning: Processing stages of the hippocampal system. Biol Cybern 86:15–28.

Goldman-Rakic PS. 1995. Cellular basis of working memory. Neuron 14:477–485.

Griffin AL, Eichenbaum H, Hasselmo ME. 2007. Spatial representations of hippocampal CA1 neurons are modulated by behavioral context in a hippocampus-dependent memory task. J Neurosci 27:2416–2423.

Hampson RE, Deadwyler SA. 1996. Ensemble codes involving hippocampal neurons are at risk during delayed performance tests. Proc Natl Acad Sci USA 93:13487–13493.

Hasselmo ME. 2005. A model of prefrontal cortical mechanisms for goal directed behavior. J Cogn Neurosci 17:1115–1129.

Hasselmo ME, Eichenbaum H. 2005. Hippocampal mechanisms for the context-dependent retrieval of episodes. Neural Network 18:1172–1190.

Holscher C, Rolls ET, Xiang J. 2003. Perirhinal cortex neuronal activity related to long-term familiarity memory in the macaque. Eur J Neurosci 18:2037–2046.

Houk JC, Adams JL, Barto AG. 1995. A model of how the basal ganglia generate and use neural signals that predict reinforcement. In Houk JC, Davis JL, Beiser DG, editors. Models of Information Processing in the Basal Ganglia. Cambridge, MA: MIT Press. pp 249–270.

Howard MW, Kahana MJ. 2002. A distributed representation of temporal context. J Math Psychol 46:269–299.

Hummel JE, Holyoak KJ. 1997. Distributed representations of structure: A theory of analogical access and mapping. Psychol Rev 104:427–466.

Jackson JC, Johnson A, Redish AD. 2006. Hippocampal sharp waves and reactivation during awake states depend on repeated sequential experience. J Neurosci 26:12415–12426.

Johnson A, Redish AD. 2005. Hippocampal replay contributes to within session learning in a temporal difference reinforcement learning model. Neural Network 18:1163–1171.

Kirwan CB, Gilbert PE, Kesner RP. 2005. The role of the hippocampus in the retrieval of a spatial location. Neurobiol Learn Mem 83:65–71.

Koene RA, Hasselmo ME. 2005. An integrate and fire model of prefrontal cortex neuronal activity during performance of goal-directed decision making. Cereb Cortex 15:1964–1981.

Krawczyk DC. 2002. Contributions of the prefrontal cortex to the neural basis of human decision making. Neurosci Biobehav Rev 26:631–664.

Lee I, Griffin AL, Zilli EA, Eichenbaum H, Hasselmo ME. 2006. Gradual translocation of spatial correlates of neuronal firing in the hippocampus toward prospective reward locations. Neuron 51:639–650.

Lenck-Santini PP, Save E, Poucet B. 2001. Place-cell firing does not depend on the direction of turn in a Y-maze alternation task. Eur J Neurosci 13:1055–1058.

Lin L-J, Mitchell TM. 1992. Memory approaches to reinforcement learning in non-Markovian domains. Tech Rep CS-92-138, Carnegie Mellon University.

Marr D. 1971. Simple memory: A theory for archicortex. Philos Trans R Soc Lond B Biol Sci 262:23–81.

McCallum AK. 1995. Reinforcement learning with selective perception and hidden state, PhD Thesis, University of Rochester.

McGaughy J, Koene RA, Eichenbaum H, Hasselmo ME. 2005. Cholinergic deafferentation of the entorhinal cortex in rats impairs encoding of novel but not familiar stimuli in a delayed nonmatch-to-sample task. J Neurosci 25:10273–10281.

Miller EK, Li L, Desimone R. 1993. Activity of neurons in anterior inferior temporal cortex during a short-term memory task. J Neurosci 13:1460–1478.

Miller EK, Erickson CA, Desimone R. 1996. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. J Neurosci 16:5154–5167.

Montague PR, Hyman SE, Cohen JD. 2004. Computational roles for dopamine in behavioural control. Nature 431:760–767.

Morris RG, Frey U. 1997. Hippocampal synaptic plasticity: Role in spatial learning or the automatic recording of attended experience? Philos Trans R Soc Lond B Biol Sci 352:1489–1503.

Morris G, Nevet A, Arkadir D, Vaadia E, Bergman H. 2006. Midbrain dopamine neurons encode decisions for future action. Nat Neurosci 9:1057–1063.

O'Reilly RC, McClelland JL. 1994. Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off. Hippocampus 4:661–682.

O'Reilly RC, Frank MJ. 2006. Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. Neural Comput 18:283–328.

Rapp PR, Amaral DG. 1989. Evidence for task-dependent memory dysfunction in the aged monkey. J Neurosci 9:3568–3576.

Redgrave P, Gurney K. 2006. The short-latency dopamine signal: A role in discovering novel actions? Nat Rev Neurosci 7:967–975.

Redish AD. 2001. The hippocampal debate: Are we asking the right questions? Behav Brain Res 127:81–98.

Riches IP, Wilson FA, Brown MW. 1991. The effects of visual stimulation and memory on neurons of the hippocampal formation and the neighboring parahippocampal gyrus and inferior temporal cortex of the primate. J Neurosci 11:1763–1779.

Schoenbaum G, Setlow B. 2001. Integrating orbitofrontal cortex into prefrontal theory: Common processing themes across species and subdivisions. Learn Mem 8:134–147.

Schultz W, Dayan P, Montague PR. 1997. A neural substrate of prediction and reward. Science 275:1593–1599.

Singh SP, Sutton RS. 1996. Reinforcement learning with replacing eligibility traces. Mach Learn 22:123–158.

Stern CE, Sherman SJ, Kirchhoff BA, Hasselmo ME. 2001. Medial temporal and prefrontal contributions to working memory tasks with novel and familiar stimuli. Hippocampus 11:337–346.

Sutton RS. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, pp 216–224.

Sutton RS, Barto AG. 1998. Reinforcement Learning: An Introduction. Cambridge: MIT Press.

Sutton RS, Precup D, Singh S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artif Intell 112:181–211.

Tulving E. 2002. Episodic memory: From mind to brain. Annu Rev Psychol 53:1–25.

Watkins CJCH. 1989. Learning from delayed rewards, PhD Thesis, Psychology Department, University of Cambridge.

Wood ER, Dudchenko PA, Eichenbaum H. 1999. The global record of memory in hippocampal neuronal activity. Nature 397:613–616.

Wood ER, Dudchenko PA, Robitsek RJ, Eichenbaum H. 2000. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. Neuron 27:623–633.

Young BJ, Otto T, Fox GD, Eichenbaum H. 1997. Memory representation within the parahippocampal region. J Neurosci 17:5183–5195.

Zilli EA, Hasselmo ME. 2005. A model of memory-guided behavior based on prefrontal cortex action selection and hippocampal episodic retrieval. Program No. 775.7. Society for Neuroscience.

Zironi I, Iacovelli P, Aicardi G, Liu P, Bilkey DK. 2001. Prefrontal cortex lesions augment the location-related firing properties of area TE/perirhinal cortex neurons in a working memory task. Cereb Cortex 11:1093–1100.

APPENDIX

Episodic Memory Equations

Here, we describe the abstract episodic memory system with equations of vectors and matrices, providing one possible way that the system might be implemented.

Let **T** be a binary row vector with n elements, where n is the number of temporal indices to be used, letting **T** have only one nonzero entry at any time. Let **S** be a binary row vector with a number of elements equal to the number of state components for each of the state elements (i.e., all of the factored state elements are treated as vectors concatenated together, the length of which is $|S_1| + |S_2| + \cdots + |S_n|$), where all entries are 0 except those representing the active states. As an example, suppose $S = \{S_{\text{vision}}, S_{\text{tactile}}\}$ where the S_{vision} element has a red, a green, and a blue component and S_{tactile} has a rough and a smooth component. If the current state is green and smooth, then, as a vector, $\mathbf{S} = [0\ 1\ 0\ 0\ 1]$. We can break **S** into separate vectors for each nonzero element, where $\mathbf{S}_i$ is a vector the same size as **S** where the ith element equals 1 and the others are zero. That is, the example **S** from above can be broken into $\mathbf{S}_2 = [0\ 1\ 0\ 0\ 0]$ and $\mathbf{S_5} = [0\ 0\ 0\ 0\ 1]$. (These vectors $\mathbf{S}_i$ should not be confused with the factored state elements S_i, where S_i is the set of values that the ith element in the state can take, i.e., the components of S_i).

We can now define matrices for associating states with temporal indices and vice versa. Let M be a matrix where the ith row contains the state **S** last associated with temporal index i. This matrix will allow retrieval of a sensory vector given a temporal index vector. Then for encoding, M is updated after each selected movement action as:

$$dM = \mathbf{T}^T\mathbf{S} - \mathbf{T}^T\mathbf{T}M$$

or

$$dM = \mathbf{T}^T(\mathbf{S} - \mathbf{T}M) \tag{A1}$$

Notice that this equation simply adds **S** to the row indicated by temporal index vector **T** (containing a single, nonzero entry) and subtracts the old memory vector, **T**M, that was stored in that temporally indexed row at some previous time. It follows that we can retrieve the sensory vector associated with **T** by using the same product:

$$\mathbf{S} = \mathbf{T}M \tag{A2}$$

This can be understood from a neural network standpoint if M is treated as a synaptic weight matrix, **S** is a vector of postsynaptic activity, and **T** a vector of presynaptic activity (here with only one presynaptic unit active). With this interpretation, Eq. (A1) becomes a learning rule with two components: $\mathbf{T}^T\mathbf{S}$ is a Hebbian term that associates the sensory vector to the temporal index vector. The second term then represents presynaptically gated long-term depression (LTD) that can be written in the simple form $\mathbf{T}^T\mathbf{T}M$ because **T** has only a single active unit (a rule analogous to A3 below can handle the general case with multiple active units). This produces a matrix the same size as M where only the ith row is nonzero (equal to **T**M) where i is the index of the nonzero unit in **T**. The equation as a whole then simultaneously applies LTD to all synapses from unit i in **T** to set them to zero except those where the postsynaptic units **S** are active.

Next, we define the matrix N with one row for each state element that contains the temporal index vector for the time the given state was last experienced. This will allow a sensory vector to retrieve an associated temporal index vector. N is updated as

$$dN = \mathbf{S}^T\mathbf{T} - \sum_i \mathbf{S}_i^T\mathbf{S}_i N \tag{A3}$$

where the summation index i is over all nonzero elements in **S** and $\mathbf{S}_i$ is a vector the same size as **S** with all entries zero except for the ith.

The neural network interpretation of this equation is identical to that for Eq. (A1) except this allows for the case of multiple active presynaptic units, by simply computing the LTD component separately for each active presynaptic unit in the summation over i.

To later retrieve the temporal index vector given a cue vector $\mathbf{S}_\text{i}$, we multiply

$$\mathbf{T} = \mathbf{S}_i N \tag{A4}$$

This can again be interpreted as a synaptic matrix and two populations of units in a neural network.

To finish characterizing the episodic memory abstraction in terms of matrices, we can define a shift operator matrix that takes a temporal index vector and advances the single nonzero entry by one position. This matrix, O, is simply an identity matrix with each row shifted up one and with the first row moved to the last row. As an example, the 3 × 3 shift operator matrix would be

$$O = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{A5}$$

So, for example, $[1\ 0\ 0]O = [0\ 1\ 0]$.

In actual implementations, two other concerns must be addressed. If the update to M is performed upon entering a state, then any cued retrieval actions in that state will retrieve the current temporal index, which was just associated to the location. It is, however, desired that the cuing retrieves the previous visit to the present state, not the current visit, so M should only be updated upon leaving a state (in the same manner that state and action values in temporal difference learning algorithms are updated for the state that was just left, not the upcoming state). Also, the agent should not be allowed to advance the temporal index past the current temporal index. In our implementation, attempts to do so return the "nothing recalled" state. Thus, the computations performed when the cuing retrieval action is taken on cue vector $\mathbf{S}_i$ are

$$\mathbf{T} = \mathbf{S}_i N$$

then

$$\mathbf{S} = \mathbf{T} M$$

On taking the advance retrieval action, we compute:

$$\mathbf{T}_{\text{new}} = \mathbf{T}_{\text{old}} O$$

then:

$$\mathbf{S} = \mathbf{T}_{\text{new}} M$$

This implementation of the model can produce retrieval errors under certain conditions. If the agent has not visited a given state in a large number of trials, it may be that the temporal index last associated with that state has been reused and no longer indexes the correct memory. Including a general decay term in the associations from states to temporal indices and thresholding retrieval may solve this problem, for example, replacing Eq. (A3) with $dN = \mathbf{S}^T\mathbf{T} - \sum_i \mathbf{S}_i^T \mathbf{S}_i N - \varepsilon N$ with $0 \leq \varepsilon \leq 1$.

Although not presented here, we have also simulated spatial memory with a nearly identical set of equations. The primary difference is that, in spatial memory, the analog of advancing retrieval is imagined motion in a particular direction. Thus, instead of a single operator matrix to advance one step in time, there are multiple operator matrices that each advance the spatial index in a different direction (e.g., in the case of four spatial directions there would be operator matrices for each of north, south, east, and west). It is also simple to take into account head or eye direction dependence of retrieved memories by including the direction as part of the spatial index (e.g., retrieving based on looking north or south from the same position would result in different states being retrieved and one would also have to include operator matrices for changing the head or eye direction without changing the location aspect of the spatial index). The spatial memory version would allow an agent to use a cue stimulus to retrieve a spatial location and sensory stimuli present there, then to mentally navigate through the spatial memory.

Multiple Retrieval Cues

The retrieval system described earlier has at least one major limitation: a given state cue might belong to multiple episodic memories and it may not be the most recent one whose retrieval is desired. To overcome this limitation, we can allow retrieval based on multiple cues so that the most recent episode containing all of the cues is retrieved, ignoring episodes containing only a subset of the cues. First, we change the updating of N so that it is no longer a binary process:

$$dN = \mathbf{S}^T\mathbf{T} - \lambda \sum_i \mathbf{S}_i^T \mathbf{S}_i N \qquad \text{(A6)}$$

where $0 < \lambda < 1$ is a decay term (if $\lambda = 1$ this equation reduces to the binary case described above). Now a given state element will be associated with multiple temporal indices with the most recent association having a value of 1, the next most recent having a value of $(1 - \lambda)$, the next most recent $(1 - \lambda)^2$ and, in general, the ith most recent having a value of $(1 - \lambda)^{i-1}$. This concept is similar in some ways to the temporal context model proposed by Howard and Kahana (2002) (also used in Hasselmo and Eichenbaum, 2005). The neural network interpretation also applies here. The only difference from Eq. (A3) is that there is a learning rate associated with the presynaptically gated LTD that is less than 1, so synapses are not completely zeroed after a single application of the learning rule.

Now given a state vector $\mathbf{S}$ with all elements 0 except i_1, i_2, . . ., i_n, we compute

$$\mathbf{T} = \left[\prod_i \mathbf{S}_i N\right]_{\max} \qquad \text{(A7)}$$

where the product is an entry-wise product of each of the vectors, $[]_{\max}$ is an operation that sets all elements to 0 except the maximum element in the vector, and $\mathbf{S}_i$ is the cue vector $\mathbf{S}$ broken into independent components as mentioned earlier. Any temporal index that is not shared by all of the $\mathbf{S}_i$ will be 0 by the multiplication, and, if more than one index is shared by all the retrieval cues, all except the most recent index will be set to 0 by the max operation. If no temporal index is shared by all of the cues, the 0 vector will result and S_{EP} is set to the "no recall" state.

To see that this works, consider each $\mathbf{T}_i = \mathbf{S}_i N$. $\mathbf{T}_i$ is a vector of the temporal indices at which a given state element has been present, weighted exponentially as described earlier such that the more recent indices have higher values. For example, suppose state element $\mathbf{S}_\text{a}$ was present at temporal indices 7, 5, and 1. The value for $\mathbf{S}_\text{a}$ at index 5 is greater than that at 1 (by a factor of $1 - \lambda$, in fact) because 5 was more recently experienced. Calculating $\mathbf{S}_\text{a}N$ will produce the first row of N as a vector. Similar calculations of $\mathbf{S}_\text{b}N$ and $\mathbf{S}_\text{c}N$ produce rows two and three as vectors. If we desire the temporal index at which all three of $\mathbf{S}_\text{a}$, $\mathbf{S}_\text{b}$, and $\mathbf{S}_\text{c}$ were most recently present, we can first find any temporal indices for which all were present, then find which of those was the most recent.

For a given temporal index, t, we know that at least one of the state elements was not present if at least one of them has a value of 0 for that index. Therefore, if we multiply the value of the association between each state element and temporal index t, the result will be 0 if at least one was not present and will be greater than 0 otherwise.

Finally, we ask which of the temporal indices shared by all elements was the most recent. Clearly it is the one for which the product of the temporal index strengths is largest. Consider two distinct, nonzero products, p and q, of temporal index strengths. Each element in N is of the form $(1 - \lambda)^z$ for some z, and each of p and q are the product of m such terms, so we have $p = (1 - \lambda)^{p_1+p_2+\cdots+p_m}$ and $q = (1 - \lambda)^{q_1+q_2+\cdots+q_m}$. If we assume without loss of generality that p corresponds to a time that happened before q (by assumption they are distinct and so do not represent the same time), then we must have $p_i > q_i$ for all i. Then the sum of all p_i must exceed the sum of all q_i so $p < q$ because $(1 - \lambda) < 1$. Thus, the largest entry in the entry-wise product of the rows from N corresponds to the most recent cooccurrence of all the state elements of interest.

Similar equations can be applied to spatial memory, as a given sensory state may be associated with multiple spatial locations.