Chapter 14: Memory systems

Memory is computationally accessible information about the past. This chapter argues that the brain stores different forms of memory in order to provide state representations appropriate for different kinds of tasks. In particular, the partial observability of the environment state necessitates computations of (approximate) belief states—posterior distributions over hidden states—which are functions of sensory history. The belief states depend on the task-specific structure of partial observability. In some cases they require only stable short-term maintenance, while in others they require temporal dynamics or long-term passive storage. This allows us to understand the logic underlying the multiplicity of memory systems in the brain.

Most neuroscience textbooks treat "memory" as a set of dedicated systems specialized for different kinds of memoranda (Figure 1). To some, extent this treatment is correct: there are areas of the brain with neurons that activate in response to storage and retrieval of particular information. Damage to these areas results in selective deficits for that information. Damage to the hippocampus, for example, selectively impairs episodic memory, leaving semantic and procedural information mostly (though not entirely) intact.

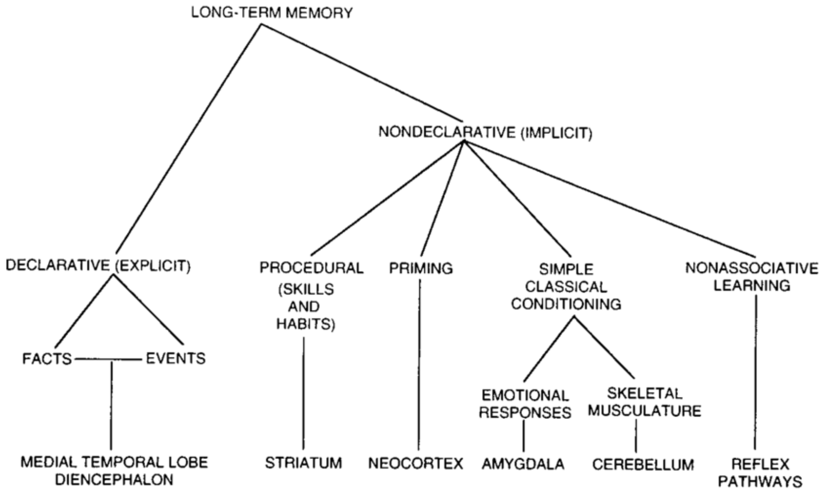

Figure 1: **A standard taxonomy of memory systems.** Reproduced from Squire and Zola (1996).

The main problem with this treatment is that it misses the pervasiveness of memory for many different computations (Dasgupta and Gershman, 2021). This chapter will take a broader view, exploring the idea that memory is fundamentally about keeping track of *state*—the information about the past that is required to predict or control the future. Which state needs to be tracked depends on the structure of the process being predicted/controlled—hence the existence of multiple brain systems encoding distinct kinds of memory.

1 *Belief states and partial observability*

To illustrate the general point that memory is everywhere, we'll start with a problem that (at first glance) doesn't seem to involve memory at all. Returning to the material from Chapter 10, let's consider the problem posed by a standard Pavlovian protocol (delay conditioning). On each trial, an animal is presented with a conditioned stimulus (CS, such as an odor, light, or tone), followed after a delay by an unconditioned stimulus (US, such as water, food, or shock). With repeated trials, the animal begins to produce a conditioned response to the CS, which reports its expectation about the upcoming US. For example, a water US will evoke in rodents anticipatory licking of the water spout. To predict the next US, the animal needs to know whether it is currently in the interstimulus interval (ISI) or the intertrial interval (ITI), as well as how much time has elapsed since the start of the interval. We will refer to the interval type (ISI or ITI) as the *macrostate* and the interval duration as the *microstate* (Figure 2). Together, these define a state space sufficient for reward prediction.

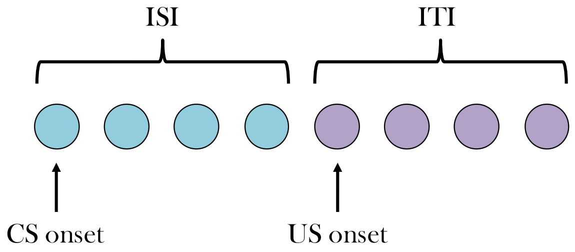

Figure 2: **The state space of a Pavlovian protocol.** Each node represents a temporal microstate (discretized here for visualization), grouped by color according to the superordinate macrostate (ISI or ITI).

To see why, note that we can write down Markov transition $T(s'|s,a)$ and reward $R(s)$ functions when $s = (m, \tau)$, with macrostate m and microstate τ. Recall that the Markov property means that the distribution over transition and rewards depends only on the current state. Formally, if H denotes the state and action history, the Markov property states that $p(s'|H) = T(s'|s,a)$ and $R(H) = R(s)$. It is in fact the definition of what it means for some set of variables to constitute a state—i.e., the set is a state if the Markov property is satisfied. When sensory observations x are sufficient to unambiguously specify the state, we say that the environment is *fully observable*.

This is sometimes known as a *semi-Markov* model, which factors the transition distribution into macrostate and microstate components. It is equivalent to a Markov model defined over the joint macro/micro space.

Note that the state space is not necessarily unique, though computationally it is useful to prefer the *minimal state* (the smallest set satisfying the Markov property).

Unfortunately, the real world, and even simple Pavlovian protocols, are rarely fully observable. The microstate representation assumes that animals are perfect time-keepers, but it's well known that their temporal precision decreases with elapsed time (see Chapter 10). This means that the microstate is *partially observable* in the sense that the sensory observations provide ambiguous evidence about elapsed time. To make matters worse, some Pavlovian protocols use partial reinforcement schedules, where the US is delivered stochastically. When combined with the partial observability of the microstate, this means that the macrostate is also partially observable: on trials when the US is omitted, the animals doesn't know for sure if it's in the ISI or the ITI. Under partial observability, sensory observations do not in general satisfy the Markov property, $p(s'|X) \neq T(s'|x,a)$, where X is the history of observations. They are thus not proper states.

The significance of partial observability lies in the fact that the Markov property is crucial to the design of efficient learning algorithms. Recall from Chapter 10 that the temporal difference (TD) learning algorithm was derived from the Bellman equation, a recursive decomposition of the value function that relies on the Markov property. In fact, many computational algorithms rely on a notion of state. For example, the Kalman filtering algorithm described in Chapter 10 also requires that its inputs satisfy an analogous Markov property. Models of decision making likewise assume, implicitly or explicitly, some summary statistic of experience that suffices to specify preferences; decisions would not be feasible if they depended on an agent's entire history.

Fortunately, there is a way to restore the Markov property for settings like Pavlovian protocols, by computing the posterior over states given the history of observations, $b(s) = p(s|X)$, or what we will call a *belief state*. As the name suggests, the belief state is truly a state—it satisfies the Markov property. In other words, the belief state is a sufficient statistic for the observation history. This means we can still use algorithms like TD learning even in partially observable environments, as long as they operate over belief states (or some approximation of belief states, as discussed below).

Before exploring the empirical implications of these ideas, let's take a step back and consider the big picture. We've posited a system for Pavlovian reward prediction that requires a particular form of history representation (i.e., a particular form of memory) encoded in the belief state. On this view, Bayesian inference is a mechanism for translating experience into memory. This is a normative prescription for how memories should be designed in order to serve the computational functions described in earlier chapters. The formal setup can be applied far beyond Pavlovian conditioning, by analyzing what the underlying state space is and then constructing the Bayesian belief state for that space. Even in cases where this computation is intractable, it offers a powerful conceptual framework for understanding the computational logic of memory.

2 *Belief states in the dopamine system*

The previous section claimed that belief states resurrect the viability of TD learning in the face of partial observability. If the brain uses this strategy, then we should expect to see signatures of belief states in phasic dopamine activity, thought to report the prediction error used by TD learning (see Chapter 10). Daw et al. (2006) suggested that this could explain a puzzling phenomenon reported by Hollerman and Schultz (1998). Animals were trained with a regular ISI and then tested with irregular ISIs. The key finding was that early reward (US) deliveries produced a pronounced excitation at the unexpected time of reward, yet no subsequent dip at the expected time of reward. This is puzzling because it would seem that this should produce a negative prediction error. However, it makes sense if one takes belief states into account. When the early reward occurs, the belief state should shift probability mass to the ITI state, during which no reward is expected.

A more direct test of belief states in the dopamine system was undertaken by Starkweather et al. (2017). The experimental design, shown in Figure 3, consisted of two different tasks (run in separate mice). These were identical except for one crucial difference: Task 1 was fully observable (no reward omissions), whereas Task 2 was partially observable (reward was omitted on 10% of trials). Mice in each task experienced trials with several different odor CSs; we will focus on Odor A, which was associated with a stochastic delay (Gaussian-distributed with a mean of 2 seconds).

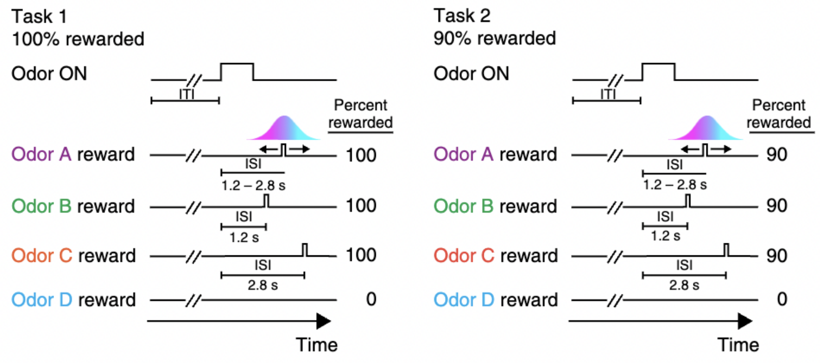

Figure 3: **Experimental design used in Starkweather et al. (2017).**

In the fully observable Task 1, the reward will always arrive, so the animal's reward expectation should grow as time elapses. The corollary of this growth is that when the reward arrives, the prediction error should be smaller for later rewards (since the prediction error is the difference between the received reward and its expectation). The situation is quite different in the partially observable Task 2: the reward may never arrive, so the animal's reward expectation should actually shrink as time elapses, due to the increasing probability that it's currently in an omission trial. Accordingly, the prediction error should be larger for later rewards. The response of dopamine neurons to reward followed this pattern, decreasing with ISI in Task 1 and increasing with ISI in task 2 (Figure 4).

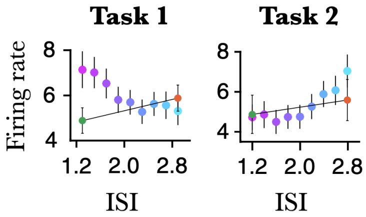

Figure 4: **Firing rate (after baseline subtraction) of dopamine neurons in response to reward after variable ISIs.** The circles connected by lines correspond to Odors B and C. Adapted from Starkweather et al. (2017).

Another source of evidence for belief states in the dopamine system comes from a study in which mice were trained to expect two different reward magnitudes (water volumes), clustered in blocks of trials (Figure 5). Here the rewards themselves are the relevant sensory signal for the hidden state. After training, the mice were tested on intermediate reward magnitudes.

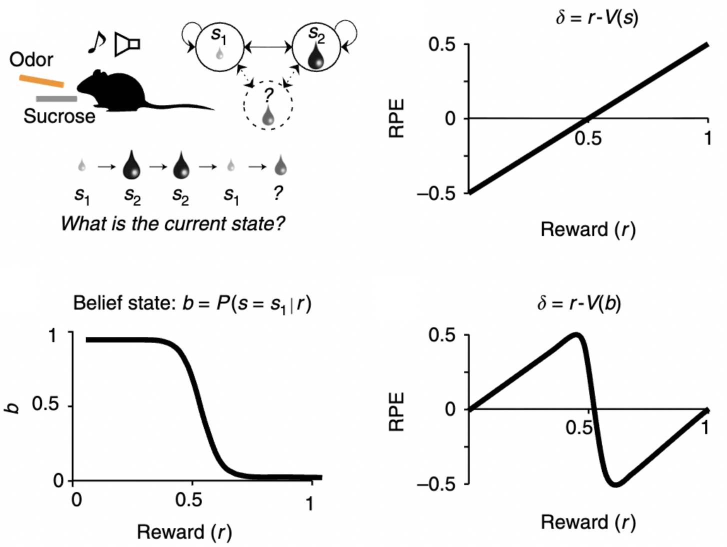

Figure 5: **Experimental design and theoretical predictions from Babayan et al. (2018).**

According to the belief state model, prediction errors (and hence dopamine responses) should grow with reward magnitude for relatively small magnitudes. This is because the "small reward" state is more probable for these magnitudes, and thus mice are receiving more than expected. The pattern should switch diametrically for relatively large magnitudes, as the "large reward" state becomes more probable and the mice are receiving less than expected. Measurements of dopamine neuron activity confirmed these predictions, exhibiting a "zig-zag" pattern as a function of reward magnitude (Figure 6).

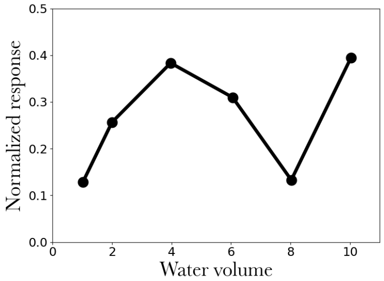

Figure 6: **Normalized responses of dopamine neurons to intermediate reward magnitudes**. Adapted from Babayan et al. (2018).

3 *Emergent belief states*

The evidence for belief states in the dopamine system comes with a significant caveat. It's only feasible to feed belief states into the TD learning machinery when these are relatively low-dimensional; in high dimensions, the brain can't compute belief states exactly (see Chapter 5). Even if it could, it might not be a good idea to run TD learning on such a high-dimensional representation due to concerns about overfitting, which would lead to poor generalization.

One strategy is to reduce belief states to a smaller set of "belief points" (Rao, 2010). Another strategy is to build a sufficiently expressive function approximator that can be trained to compute representations suitable for reward prediction (Ni et al., 2022). Because belief states are sufficient for reward prediction, this could plausibly yield belief-like representations. However, this is not guaranteed, since belief states are not the only valid choice of state representation; in some cases, much more compressed representations are possible (Roy et al., 2005), which might not even resemble belief states.

This strategy is closely related to the Monte Carlo approximations discussed in Chapter 5.

Hennig et al. (2023) studied a recurrent neural network (RNN) as the function approximator for value. Unlike the linear function approximators discussed in Chapters 10 and 11, the Value RNN is able to learn new representations of sensory data. By examining the structure of these representations, Hennig et al. showed that they are "belief-like" in the sense that their dynamics resembled what we would expect from belief state updating (see next section), and belief states could be approximately decoded from them (Figure 7). Importantly, the Value RNN representations were lower dimensional than the belief states, retaining only the information in memory needed to predict reward. Thus, compressed, belief-like representations can be an emergent property of learning. Finally, Hennig et al. showed that prediction errors generated by the Value RNN qualitatively capture the patterns of dopamine neuron activity reviewed in the last section, establishing that exact belief states are not necessary.

This fits with anatomical arguments that the striatum acts as a reward-driven representational bottleneck, dramatically compressing cortical inputs (Bar-Gad et al., 2003).

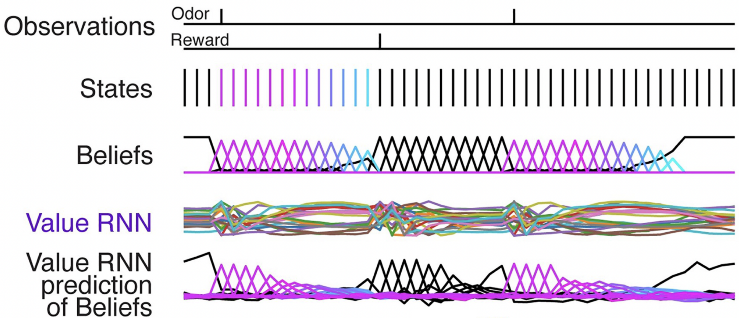

Figure 7: **Belief state decoding from a recurrent neural network.** The Value RNN is a recurrent neural network trained to predict future reward (i.e., to estimate value). The fourth row shows the time course of units in the Value RNN. The fifth row shows belief states decoded linearly from the Value RNN units. Colors correspond to temporal microstates, as in Figure 4. Reproduced from Hennig et al. (2023).

Taking stock of the story so far, we started with the problem of partial observability, arguing that memory is essentially a quest for state: efficient learning, prediction, and decision making all rely on some form of Markov property, often violated in partially observable environments, and restorable by an appropriate choice of memory. Bayesian belief states offer a principled choice of memory, but we showed that this is not necessary: an RNN trained to predict reward also acquired the appropriate memory, a compressed version of belief states. This suggests a general strategy the brain might use to solve the problem of partial observability.

4 *Recurrent neural network models of working memory*

A closer look at the population dynamics of the Value RNN is instructive. For visualization purposes, it's useful to project the population activity into a 2D subspace spanned by the first two principal components (Figure 8). Principal components analysis (PCA) is a linear dimensionality reduction method, which can be obtained from the singular value decomposition $X = UWB^\top$, where (in our setting) X is the time-by-neuron activity matrix, U is a time-by-time matrix (the left singular vectors), W is a time-by-neuron diagonal matrix (the singular values) in decreasing order, and B is a neuron-by-neuron matrix (the right singular vectors). The top K principal components are obtained by taking the first K columns of the matrix formed by UW. These are the components that capture the most variance in the data, in the sense that reconstructing the data using only these components will give the best approximation of all possible combinations of components.

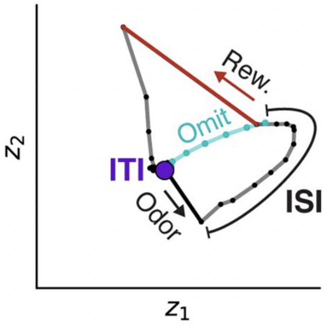

Figure 8: **Dynamics of the Value RNN trained on Task 2 of Starkweather et al. (2017).** Each axis represents one principal component. Reproduced from Hennig et al. (2023).

This analysis reveals a single "fixed point" corresponding to the ITI, which is stable in the absence of sensory input. Such fixed point structure is rational for the Starkweather task, because the ITI is drawn from a geometric distribution, which has the property that elapsed time in the ITI is uninformative about the time until the next trial. When an odor is presented, the activity is abruptly kicked out of the fixed point, and then continues to slowly evolve after the odor is removed, until reward is received (which kicks the system into another part of the activity space, gradually decaying back to the ITI fixed point) or the next trial begins (on omission trials).

The dynamics exhibit both memory maintenance (the trajectory encodes a trace of the stimulus) and time-keeping (the trajectory encodes a representation of elapsed time). This reflects the underlying state structure: the stimulus trace corresponds to the macrostate, and the temporal trace corresponds to the microstate. It turns out that a similar mixing of stimulus and timing information is present in more conventional “working memory” tasks involving the short-term maintenance and manipulation of stimulus information.

In a classic study, Fuster and Alexander (1971) recorded neurons in prefrontal cortex while monkeys performed a delayed selection task. A food was hidden underneath one of two objects, followed by a delay period during which the monkeys could not see the objects. After the delay period, monkeys could then retrieve the food. Some neurons were activated by the cue (object presentation), and then exhibited persistent firing during the delay interval (Figure 9).

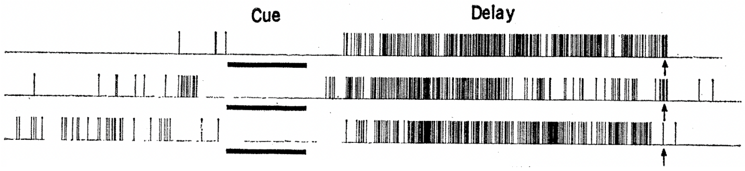

Figure 9: **Spikes recorded from a prefrontal neuron during a delayed selection task**. Reproduced from Fuster and Alexander (1971).

This discovery heralded many subsequent studies of persistent activity during working memory tasks, as well as RNN models designed to explain this phenomenon (see Wang, 2021, for a review). A standard mechanism in these models is recurrent excitation that produces “reverberating” activity, allowing stimulus representation to be maintained even after the stimulus has disappeared. To store information persistently, the network must represent the stimulus as an *attractor*—a network state that is robust to small perturbations. We already encountered a particular kind of attractor (a fixed point) in our discussion of the Value RNN dynamics. Attractors can also be lines and rings (continua of stable states with different topologies), sometimes exhibiting oscillatory dynamics.

Some form of inhibition is also usually needed to prevent runaway excitation.

4.1 *Beyond persistent activity*

Persistent activity is usually understood to mean the kind of pattern noted by Fuster and Alexander: single neurons that are activated by a stimulus and continue firing during a delay period. Note, however, that persistence in the informational sense (the ability to read out stimulus information from neural activity) does not inherently require persistent activity in the classical sense. In fact, typically only a small proportion (5-10%) of recorded prefrontal neurons satisfy the classical definition. Many neurons have complex time-varying responses, including ramps and oscillations. How, then, does the prefrontal cortex stably maintain information in memory?

This brings us back to the analysis of population activity. Murray et al. (2017) used PCA to characterize prefrontal activity recorded during working memory tasks (the oculomotor delayed response, or ODR, task is illustrated in Figure 10). They did this in a somewhat

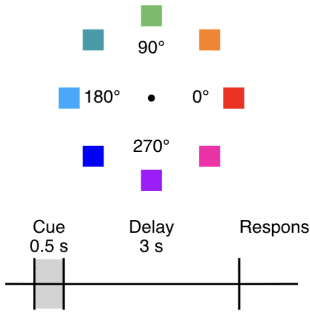

Figure 10: **The oculomotor delayed response (ODR) task.** On each trial, the subject fixates at the center while a cue is presented at one 8 locations. After a delay, the subject makes a saccade to the remembered location. Reproduced from Murray et al. (2017), based on the task studied in Constantinidis et al. (2001).

different way than what was described above. First, they constructed a stimulus-by-neuron matrix by averaging delay activity for each stimulus-neuron pair; this discards time information but retains stimulus information, so they referred to the top principal components as the *mnemonic subspace*. They then did a complementary analysis of the time-by-neuron matrix obtained by averaging across stimuli for each time-neuron pair, yielding a *dynamic subspace* (orthogonalized with respect to the mnemonic subspace) that varies with time but discards stimulus information.

The results of the subspace analysis for the ODR task are shown in Figure 11. As expected based on the structure of the analysis, the mnemonic subspace reflects the stimulus location, while the dynamic subspace reflects orthogonal temporal variation. These results are non-trivial, because Murray et al. showed that several standard models cannot capture them. For example, they studied a "stable attractor" model that has been used to model persistent activity during ODR tasks (e.g., Compte et al., 2000). Each neuron is tuned to a particular location. Neurons tuned to nearby locations excite one another, while inhibiting neurons tuned to distant locations. This produces a ring attractor: stimulus information is stably maintained as an activity bump on a ring of locations. Small perturbations are resisted, while larger perturbations cause the bump to move around the ring. Consequently, the model does not show much temporal variation during the delay period (Figure 12), in contrast to the prefrontal population.

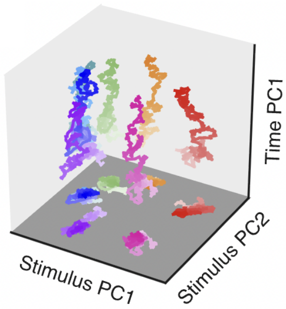

Figure 11: **Subspace analysis of prefrontal activity during the ODR task**. Population trajectories are projected into the mnemonic subspace (Stimulus PC1 and PC2) and the dynamic subspace (Time PC1), color-coded by location, as in Figure 10. Reproduced from Murray et al. (2017), based on data reported in Constantinidis et al. (2001).

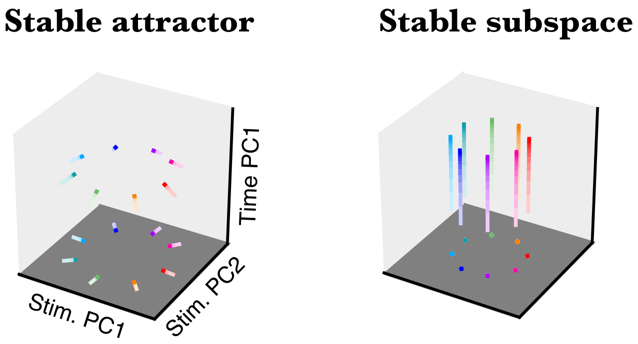

Figure 12: **Subspace analysis of two models during the ODR task**. Reproduced from Murray et al. (2017).

Motivated by their empirical results, Murray et al. (2017) developed a "stable subspace" model that is similar in many respects to the stable attractor model, but with a key difference: the network connectivity structure is organized to have a dynamic subspace in addition to the mnemonic subspace. By construction, activity varies across time in the dynamic subspace, while remaining stable in the mnemonic subspace. This reproduces the distinctive geometry of the neural data when population activity is projected onto the principal components (Figure 12).

4.2 *Modeling stable attractors and stable subspaces*

To make these ideas more formal, let's write down the firing rate dynamics of a linear dynamical system, variants of which which we've now used several times:

$$\tau \dot{x} = -x + Wx + g(s), \tag{1}$$

where τ is the time constant, W is a recurrent weight matrix, and $g(s)$ is the feedforward drive for stimulus s. In a canonical stable attractor network for a one-dimensional variable (e.g., angle in the ODR task), neuron d is tuned to a particular stimulus, which we denote by s_i^*, such that $g_i(s)$ is maximal at s_i^*. The recurrent weight matrix is specified by a symmetric kernel K that depends only on the distance between the preferred stimuli of two neurons: $W_{ij} = K(|s_i^* - s_j^*|)$. This guarantees that W is a symmetric normal matrix, with as many orthogonal eigenvectors as there are neurons; each eigenvector corresponds to one coding direction in stimulus space (aligned with the tuning function of the corresponding neuron) that can be represented independently of other coding directions. If a stimulus is presented that maximally excites neuron i, an activity bump will form at neuron i and its neighbors, corresponding to one of the eigenvectors. As long as at least one eigenvalue of W is equal to 1, the bump will be stable (a fixed point attractor). In the absence of a stimulus, any point on the ring is stable (a ring attractor).

More generally, coding M dimensions requires at least M eigenvalues equal to 1.

A stable subspace attractor can be constructed in the same way, except with a non-normal connectivity matrix. This can be accomplished by having some of the eigenvectors point in directions that are not aligned with any of the tuning functions. Rather than relaxing into a bump when stimulated, these eigenvectors will evolve over time in ways that are not strongly stimulus-driven—this is the dynamic subspace. The eigenvectors that are aligned with stimulus tuning constitute the mnemonic subspace.

4.3 *What are subspaces and why do we have them?*

From a decoding perspective, subspaces are just different choices of linear readout weights. This means that the same network can support downstream processing of both stimulus and temporal information. This multiplexing also supports more complex tasks, where stimulus and temporal information need to be combined, such as in the Starkweather task discussed earlier. More generally, the use of

Another example of multiplexing is in the anterior lateral motor cortex of mice, which encodes preparatory activity for an upcoming action (Inagaki et al., 2019). When the stimulus-response delay is fixed (but not when it's random), the activity exhibits ramping dynamics to a fixed point representing the chosen action.

multiple subspaces to encode different representations allows networks to flexibly perform a variety of tasks. This gives rise to "mixed selectivity" (tuning of the same neuron to multiple stimulus features), found in many parts of the brain, particularly the prefrontal cortex (Tye et al., 2024).

One way to look at mixed selectivity is through the lens of the memory-state duality. In many organisms, the relevant state spaces are preordained by their biological niche, and therefore they have specialized memory systems for particular kinds of memoranda. Primates (and a few other species) live in more complex and variable niches, where the relevant state space might need to be learned on the fly. This doesn't mean that primates have unlimited flexibility; task performance is still constrained by the feature space represented in areas like the prefrontal cortex. The point is that this feature space is high-dimensional, enabling greater flexibility than would be possible with the low-dimensional structure implemented in specialized memory systems.

For example, ring attractor networks are used in flies to maintain a stable memory of heading direction (Kim et al., 2017). This network is used for one and only one task.

5 *Long-term memory*

The kinds of memory we have been talking about so far are short-lived, limited fundamentally by leakage and noise—processes that can only be resisted by refreshing of the activity state encoding information. Yet somehow we are able to retain certain memories over weeks, months, and years. It is widely believed that such long-term storage depends on modification of synaptic strength, a "passive" storage medium that does not require continual refreshing (although, like other cellular processes, synapses need to be maintained in the face of molecular turnover). Before getting into biological mechanisms, let's return to the theme of this chapter, asking what kind of partial observability is being solved by a long-term memory system.

Other long-term storage mechanisms might be at play as well, such as molecular memory codes (Gershman, 2023).

Consider an agent exposed to a stream of observations, $x(t)$. To predict the next observation $x(t+1)$, it needs to know the observations at a set of past time points $\{\tilde{t}_1, \ldots, \tilde{t}_N\}$. Since these are not part of the observation at time t, the process is partially observable. Thus, to define a Markov process, we require a state representation $s(t) = (x(\tilde{t}_1), \ldots, x(\tilde{t}_N))$. What makes this hard is that the relevant temporal indices are not necessarily fixed; they can potentially change across time. For example, suppose you need to remember where you put your keys. This entails retrieving the past time point at which you last saw your keys, which is different every time you are faced with this query. The long-term memory system must therefore be able to retrieve information arbitrarily far in the past. It must also be able to answer a wide range of queries for the same observation stream.

One possible solution is a single high-capacity database of past observations. Then the problem becomes selectively retrieving the right information for a given query (i.e., reducing interference between memories). Selective retrieval is facilitated by labeling memories with addresses that can be matched to search queries, much like modern databases. As we will see, the structure of the address space is an important design question.

5.1 Retrieval interference is the critical limiting factor

You might be thinking to yourself: If long-term memory really has such high capacity, why do I forget so many things? Why can't I remember the name of my first grade teacher or what I ate for breakfast 10 years ago? Although naturally the brain, like any physical memory system, has a capacity limit, it may come as a surprise that this capacity limit (whatever it might be) is not the limiting factor on memory performance. Instead, it is interference between memories at the time of retrieval that seems to be the critical limiting factor.

The evidence supporting this argument comes from a number of different sources. First, though we lack a decisive measure of the brain's memory capacity, estimates typically range between 10^7 and $10^1$5 bits (Dudai, 1997). Estimates of total minimally compressed sensory input range between $10^1$3 and $10^1$7 bits. Thus, with more powerful compression, much of the sensory input over a lifetime could plausibly be stored in the brain.

These very loose estimates only take us so far. More convincing is evidence from behavior. If human memory ever actually hit the capacity limit, we would expect an abrupt inability to store new information without erasing old information. Yet no evidence for this prediction exists. On the contrary, studies of human memory show that information can be stored for decades despite minimal rehearsal. For example, Hall (1991) showed substantial retention of high school algebra for half a century even in the absence of rehearsal during the intervening years. Furthermore, a capacity limit explanation for forgetting would not be able to explain why we forget over much shorter timescales.

When information is forgotten, it is usually not permanently erased but only temporarily inaccessible. For example, memory performance typically declines with longer retention intervals, but careful experiments reveal that it is accessibility, not precision, that is declining (Berens et al., 2020; Diamond et al., 2020). When very old memories are retrieved, they are just as precise as recent memories.

Many studies have shown that supposedly lost memories can be recovered under a variety of circumstances, including appropriate retrieval cues (Wagenaar, 1986), more retrieval attempts (Buschke, 1974), and even just waiting long enough before attempting retrieval (Payne, 1987). Amnesia induced experimentally or from neurological damage is also frequently reversible (Lewis et al., 1968; Kapur, 1999).

This brief survey hopefully makes clear that a high-capacity long-term memory system in the brain is empirically plausible. The challenge we take up in the next section is to make it computationally plausible.

5.2 *The key-value data structure*

Many books include an alphabetical index that specifies where in the book important information can be found. The alphabetical organization is useful for information retrieval, but does not carry any information content itself. The brain may use a similar division of labor between indexing and content (Gershman et al., 2025). We will first formalize this idea using a formalism from machine learning, and then discuss how it maps onto brain circuitry.

A workhorse of modern machine learning is the transformer architecture (Vaswani et al., 2017), the foundation of sequence processing systems like large language models. Transformers use a parallelized "attention" mechanism to select past inputs for retrieval. Each input $x(t)$ is associated with a *key* vector $k(t)$ and a *value* vector $v(t)$. Intuitively, the key vector is an index that defines the "address" (analogous to page number in a book) where content, represented by the value vector, is stored. When input $x(t)$ is received, it gets mapped to a query vector $q(t)$ in the same address space as the keys, so that they can be matched by a similarity function $S(k, q)$. These similarity values are then used to compute an attention score vector $\alpha = \sigma[S(k(< t), q(t))]$, where $k(< t)$ denotes the set of keys at past time points, with the similarity function evaluated in parallel all of these time points, and $\sigma[\cdot]$ is a "separator" function that can amplify strong matches and suppress weak matches. The most widely used similarity function is the inner product: $S(k, q) = qk^\top$. The most widely used separator function is the softmax:

$$\sigma[S(k(< t), q(t))] = \frac{\exp[S(k(< t), q(t))]}{\sum_{t' < t} \exp[S(k(t'), q(t))]}. \qquad (2)$$

The resulting attention score defines the retrieval strength for content (the value vector) linked to each past input, such that the retrieved value is a linear composite of past values weighted by their attention scores:

$$\hat{v}(t) = \sum_{t' < t} \alpha(t') v(t'). \qquad (3)$$

The term "attention" here has a specific technical meaning which does not cleanly map onto the mechanisms of attention discussed in Chapter 6.

"Value" is another technical term which is unrelated to its meaning in RL theory.

The mappings from observations to keys, queries, and values, are typically learned linear transformations.

Because the attention vector is normalized, we can potentially interpret it as a belief state—i.e., posterior probabilities over a hidden state. In this case, the hidden state $s \in \{1, \dots, t-1\}$ corresponds to which past input is relevant at the current time point. The retrieved value can then be understood as the expectation of the value vectors under the posterior.

The implicit joint probability distribution over hidden state s and query q is given by $p(s, q) \propto \exp[S(k(s), q)]$. The belief state is given by $b(s) = p(s|q)$, which is equivalent to the attention score vector α.

5.3 How the brain implements key-value memory

There are multiple ways to implement key-value memory in the brain (e.g., Tyulmankov et al., 2021; Whittington et al., 2022; Kozachkov et al., 2023; Chandra et al., 2025), though we will not review them in detail. A general schema for key-value memory is shown in Figure 13. Sensory inputs project to a population of neurons encoding attention scores, which then project to an output population encoding retrieved value. The input-to-attention weights correspond to keys; the attention-to-output weights correspond to values.

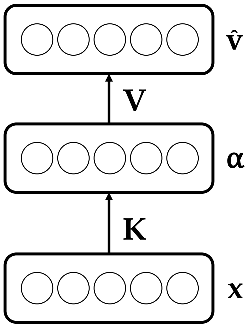

Figure 13: **Neural architecture for key-value memory**.

To a first approximation, this architecture can be mapped onto the hippocampal-cortical system, where the hippocampus encodes attention scores, which are used to retrieve content (value) stored in cortex. On this view, the hippocampus does not itself encode memory content, but rather serves as an address space for similarity-based indexing. Consistent with this view, memory retrieval relies on reinstatement of activity in cortex (putatively activation of the output layer in Figure 13), which depends on the hippocampus (Tanaka et al., 2014; Hebscher et al., 2021). When parts of cortex are degraded, for example in semantic dementia, an input can still be recognized as old vs. new (i.e., the index is relatively intact), but little semantic information about the input can be retrieved (Graham et al., 1999).

This view formalizes a venerable idea about hippocampal function, the hippocampal indexing theory (Teyler and DiScenna, 1986; Goode et al., 2020).

How is the hippocampal address space organized? One clue comes from a study of food caching birds (black-capped chickadees), who have the remarkable ability to remember hundreds of cache locations over several weeks with millimeter precision (Hitchcock and Sherry, 1990). Chettih et al. (2024) recorded from hippocampal neurons during food caching and retrieval, identifying a sparse and transient "barcode" pattern of activation that was unique to individual caches and reinstated at retrieval (Figure 14). Fang et al. (2025) showed that barcodes could be captured by an RNN with random recurrent weights, transiently amplified during memory storage to produce an effectively random pattern of activity. By associating these patterns with sensory input (the input-to-key mapping in the key-value framework), the patterns were reactivated when a cache location was approached.

The barcode acts similarly to a hash code in computer science, assigning memories to memory addresses with fixed dimension. A more sophisticated random addressing scheme was developed by Chandra et al. (2025), building on the entorhinal attractor network discussed in Chapter 13.

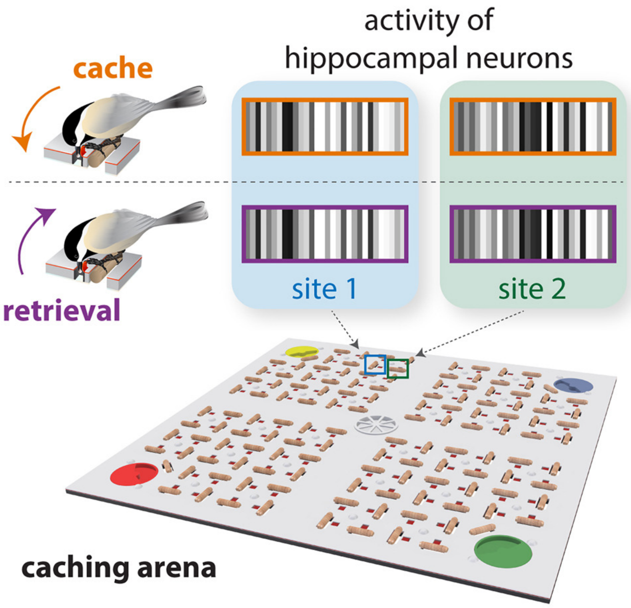

Figure 14: **Schematic of hippocampal barcode activity during food caching and retrieval**. Reproduced from Chettih et al. (2024).

6 *Conclusion*

This chapter conceptualized memory as a solution to partial observability—the unavailability of state information, which violates the Markov property and cripples efficient computation. This conceptualization helps us appreciate the logic underlying the organization of memory systems—why do we have different forms of memory at all? Different memory systems are designed to deal with different forms of partial observability, by constructing approximate belief state representations tailored to the problem domain. Belief states restore the Markov property and rescue efficient computation. In brief, *memory is a quest for state*.

Further implications of this conceptualization are discussed in Gershman and Daw (2017).

Study questions

1. The study of food caching in birds suggests a random addressing scheme in the hippocampus, much like a hash code. What are the advantages and disadvantages of such a scheme?
2. How can we reconcile the different functional views of the hippocampus (predicting, modeling, remembering) that have been discussed in this chapter and earlier chapters?

References

Babayan, B. M., Uchida, N., and Gershman, S. J. (2018). Belief state representation in the dopamine system. *Nature Communications*, 9:1891.

Bar-Gad, I., Morris, G., and Bergman, H. (2003). Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Progress in Neurobiology*, 71:439–473.

Berens, S. C., Richards, B. A., and Horner, A. J. (2020). Dissociating memory accessibility and precision in forgetting. *Nature Human Behaviour*, 4:866–877.

Buschke, H. (1974). Spontaneous remembering after recall failure. *Science*, 184:579–581.

Chandra, S., Sharma, S., Chaudhuri, R., and Fiete, I. (2025). Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638:739–751.

Chettih, S. N., Mackevicius, E. L., Hale, S., and Aronov, D. (2024). Barcoding of episodic memories in the hippocampus of a food-caching bird. *Cell*, 187:1922–1935.

Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, 10:910–923.

Constantinidis, C., Franowicz, M. N., and Goldman-Rakic, P. S. (2001). Coding specificity in cortical microcircuits: a multiple-electrode analysis of primate prefrontal cortex. *Journal of Neuroscience*, 21:3646–3655.

Dasgupta, I. and Gershman, S. J. (2021). Memory as a computational resource. *Trends in Cognitive Sciences*, 25:240–251.

Daw, N. D., Courville, A. C., and Touretzky, D. S. (2006). Representation and timing in theories of the dopamine system. *Neural Computation*, 18:1637–1677.

Diamond, N. B., Armson, M. J., and Levine, B. (2020). The truth is out there: Accuracy in recall of verifiable real-world events. *Psychological Science*, 31:1544–1556.

Dudai, Y. (1997). How big is human memory, or on being just useful enough. *Learning & Memory*, 3:341–365.

Fang, C., Lindsey, J., Abbott, L. F., Aronov, D., and Chettih, S. (2025). Barcode activity in a recurrent network model of the hippocampus enables efficient memory binding. *eLife*, 14.

Fuster, J. M. and Alexander, G. E. (1971). Neuron activity related to short-term memory. *Science*, 173:652–654.

Gershman, S. J. (2023). The molecular memory code and synaptic plasticity: A synthesis. *Biosystems*, 224:104825.

Gershman, S. J. and Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual Review of Psychology*, 68:101–128.

Gershman, S. J., Fiete, I., and Irie, K. (2025). Key-value memory in the brain. *Neuron*, 113:1694–1707.

Goode, T. D., Tanaka, K. Z., Sahay, A., and McHugh, T. J. (2020). An integrated index: engrams, place cells, and hippocampal memory. *Neuron*, 107:805–820.

Graham, K. S., Patterson, K., and Hodges, J. R. (1999). Episodic memory: new insights from the study of semantic dementia. *Current Opinion in Neurobiology*, 9:245–250.

Hall, L. K. (1991). Lifetime maintenance of high school mathematics content. *Journal of Experimental Psychology: General*, 120:20–33.

Hebscher, M., Kragel, J. E., Kahnt, T., and Voss, J. L. (2021). Enhanced reinstatement of naturalistic event memories due to hippocampal-network-targeted stimulation. *Current Biology*, 31:1428–1437.

Hennig, J. A., Romero Pinto, S. A., Yamaguchi, T., Linderman, S. W., Uchida, N., and Gershman, S. J. (2023). Emergence of belief-like representations through reinforcement learning. *PLOS Computational Biology*, 19:e1011067.

Hitchcock, C. L. and Sherry, D. F. (1990). Long-term memory for cache sites in the black-capped chickadee. *Animal Behaviour*, 40:701–712.

Hollerman, J. R. and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature Neuroscience*, 1:304–309.

Inagaki, H. K., Fontolan, L., Romani, S., and Svoboda, K. (2019). Discrete attractor dynamics underlies persistent activity in the frontal cortex. *Nature*, 566:212–217.

Kapur, N. (1999). Syndromes of retrograde amnesia: A conceptual and empirical synthesis. *Psychological Bulletin*, 125:800–825.

Kim, S. S., Rouault, H., Druckmann, S., and Jayaraman, V. (2017). Ring attractor dynamics in the Drosophila central brain. *Science*, 356:849–853.

Kozachkov, L., Kastanenka, K. V., and Krotov, D. (2023). Building transformers from neurons and astrocytes. *Proceedings of the National Academy of Sciences*, 120:e2219150120.

Lewis, D. J., Misanin, J. R., and Miller, R. R. (1968). Recovery of memory following amnesia. *Nature*, 220:704–705.

Murray, J. D., Bernacchia, A., Roy, N. A., Constantinidis, C., Romo, R., and Wang, X.-J. (2017). Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex. *Proceedings of the National Academy of Sciences*, 114:394–399.

Ni, T., Eysenbach, B., and Salakhutdinov, R. (2022). Recurrent model-free RL can be a strong baseline for many POMDPs. In *International Conference on Machine Learning*, pages 16691–16723. PMLR.

Payne, D. G. (1987). Hypermnesia and reminiscence in recall: A historical and empirical review. *Psychological Bulletin*, 101:5–27.

Rao, R. P. (2010). Decision making under uncertainty: a neural model based on partially observable Markov decision processes. *Frontiers in Computational Neuroscience*, 4:146.

Roy, N., Gordon, G., and Thrun, S. (2005). Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, 23:1–40.

Squire, L. R. and Zola, S. M. (1996). Structure and function of declarative and nondeclarative memory systems. *Proceedings of the National Academy of Sciences*, 93:13515–13522.

Starkweather, C. K., Babayan, B. M., Uchida, N., and Gershman, S. J. (2017). Dopamine reward prediction errors reflect hidden-state inference across time. *Nature Neuroscience*, 20:581–589.

Tanaka, K. Z., Pevzner, A., Hamidi, A. B., Nakazawa, Y., Graham, J., and Wiltgen, B. J. (2014). Cortical representations are reinstated by the hippocampus during memory retrieval. *Neuron*, 84:347–354.

Teyler, T. and DiScenna, P. (1986). The hippocampal memory indexing theory. *Behavioral Neuroscience*, 100:147–154.

Tye, K. M., Miller, E. K., Taschbach, F. H., Benna, M. K., Rigotti, M., and Fusi, S. (2024). Mixed selectivity: Cellular computations for complexity. *Neuron*, 112:2289–2303.

Tyulmankov, D., Fang, C., Vadaparty, A., and Yang, G. R. (2021). Biological learning in key-value memory networks. *Advances in Neural Information Processing Systems*, 34:22247–22258.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wagenaar, W. A. (1986). My memory: A study of autobiographical memory over six years. *Cognitive Psychology*, 18:225–252.

Wang, X.-J. (2021). 50 years of mnemonic persistent activity: quo vadis? *Trends in Neurosciences*, 44:888–902.

Whittington, J. C. R., Warren, J., and Behrens, T. E. (2022). Relating transformers to models and neural representations of the hippocampal formation. In *International Conference on Learning Representations*.