Lecture 14: Memory systems

Samuel Gershman

Harvard University

Roadmap

- Memory is computationally accessible information about the past.

Roadmap

- Memory is computationally accessible information about the past.
- The brain stores different forms of memory in order to provide state representations appropriate for different kinds of tasks.

Roadmap

- Memory is computationally accessible information about the past.
- The brain stores different forms of memory in order to provide state representations appropriate for different kinds of tasks.
- The partial observability of the environment state necessitates computations of (approximate) belief states—posterior distributions over hidden states—which are functions of sensory history.

Roadmap

- Memory is computationally accessible information about the past.
- The brain stores different forms of memory in order to provide state representations appropriate for different kinds of tasks.
- The partial observability of the environment state necessitates computations of (approximate) belief states—posterior distributions over hidden states—which are functions of sensory history.
- 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.

Roadmap

- Memory is computationally accessible information about the past.
- The brain stores different forms of memory in order to provide state representations appropriate for different kinds of tasks.
- The partial observability of the environment state necessitates computations of (approximate) belief states—posterior distributions over hidden states—which are functions of sensory history.
- 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.

What are memory systems?

- Neuroscience textbooks treat "memory" as a set of dedicated systems specialized for different kinds of memoranda.

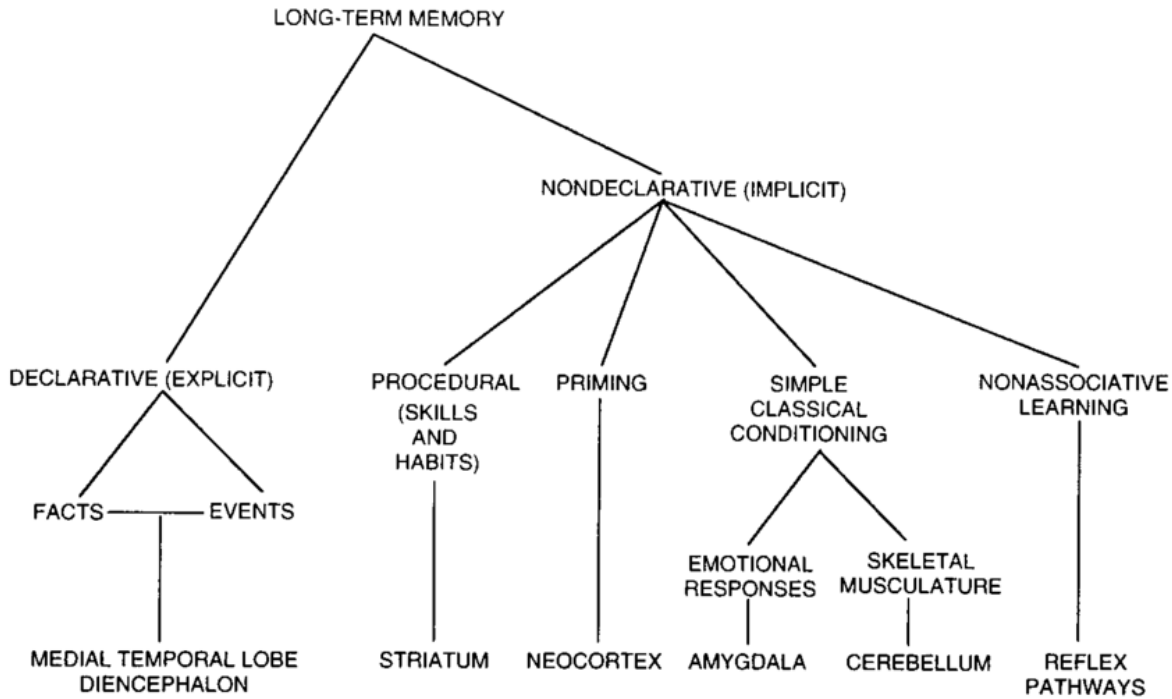
What are memory systems?

- Neuroscience textbooks treat "memory" as a set of dedicated systems specialized for different kinds of memoranda.
- Some brain areas have neurons that activate in response to storage and retrieval of particular information.

What are memory systems?

- Neuroscience textbooks treat "memory" as a set of dedicated systems specialized for different kinds of memoranda.
- Some brain areas have neurons that activate in response to storage and retrieval of particular information.
- Damage to these areas (e.g., hippocampus) results in selective deficits for that information.

Classical view of memory systems



[Squire & Zola 1996]

The normative logic of multiple memory systems

- The classical view misses the pervasiveness of memory for many different computations.

The normative logic of multiple memory systems

- The classical view misses the pervasiveness of memory for many different computations.
- We will explore 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.

The normative logic of multiple memory systems

- The classical view misses the pervasiveness of memory for many different computations.
- We will explore 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 environment—hence the existence of multiple brain systems encoding distinct kinds of memory.

Memory is everywhere: Pavlovian illustration

- To illustrate the point that memory is everywhere, we'll start with a problem that (at first glance) doesn't seem to involve memory at all: Pavlovian conditioning.

Memory is everywhere: Pavlovian illustration

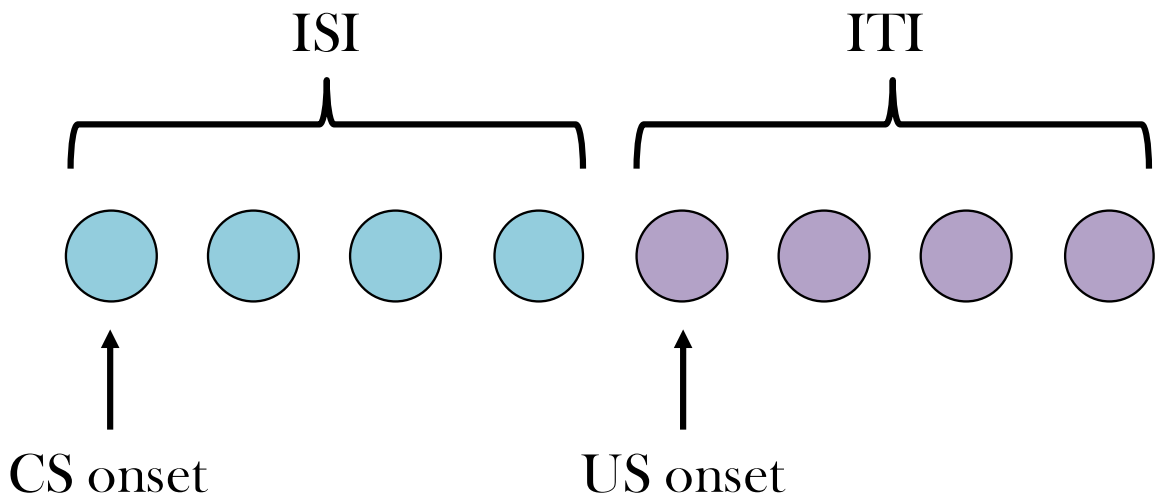
- To illustrate the point that memory is everywhere, we'll start with a problem that (at first glance) doesn't seem to involve memory at all: Pavlovian conditioning.
- To predict the next US, the animal needs to know whether it is currently in the interstimulus interval (ISI) or the intertrial interval (ITI), as well as how much time has elapsed since the start of the interval.

Memory is everywhere: Pavlovian illustration

- To illustrate the point that memory is everywhere, we'll start with a problem that (at first glance) doesn't seem to involve memory at all: Pavlovian conditioning.
- 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.
- Interval type (ISI or ITI) = *macrostate*; interval duration = *microstate*. Together, these define a state space sufficient for reward prediction.

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).



Partial observability

- The world is rarely fully observable. This is also true of the Pavlovian protocol.

Partial observability

- The world is rarely fully observable. This is also true of the Pavlovian protocol.
- The microstate representation assumes that animals are perfect time-keepers, but temporal precision decreases with elapsed time. This means that the microstate is *partially observable*: sensory observations provide ambiguous evidence about elapsed time.

Partial observability

- The world is rarely fully observable. This is also true of the Pavlovian protocol.
- The microstate representation assumes that animals are perfect time-keepers, but temporal precision decreases with elapsed time. This means that the microstate is *partially observable*: sensory observations provide ambiguous evidence about elapsed time.
- In addition, sometimes 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.

The Markov property and partial observability

- Under partial observability, sensory observations do not in general satisfy the Markov property.

The Markov property and partial observability

- Under partial observability, sensory observations do not in general satisfy the Markov property.
- The Markov property is crucial to the design of efficient learning algorithms.

The Markov property and partial observability

- Under partial observability, sensory observations do not in general satisfy the Markov property.
- The Markov property is crucial to the design of efficient learning algorithms.
- Many computational algorithms (e.g., TD learning, Kalman filtering) rely on a notion of state.

Belief states

- We can restore the Markov property by computing the posterior over states given the history of observations, $b(s) = p(s|X)$, the *belief state*.

Belief states

- We can restore the Markov property by computing the posterior over states given the history of observations, $b(s) = p(s|X)$, the *belief state*.
- 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.

Belief states

- We can restore the Markov property by computing the posterior over states given the history of observations, $b(s) = p(s|X)$, the *belief state*.
- 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.
- We can thus use algorithms like TD learning even in partially observable environments, as long as they operate over belief states (or some approximation).

Belief states and memory

- 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.

Belief states and memory

- 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.
- Bayesian inference is a mechanism for translating experience into memory—a normative prescription for how memories should be designed in order to serve the computational requirements of state-dependent algorithms.

Belief states and memory

- 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.
- Bayesian inference is a mechanism for translating experience into memory—a normative prescription for how memories should be designed in order to serve the computational requirements of state-dependent algorithms.
- Logic can be applied beyond Pavlovian conditioning, by analyzing what the underlying state space is and then constructing the Bayesian belief state for that space.

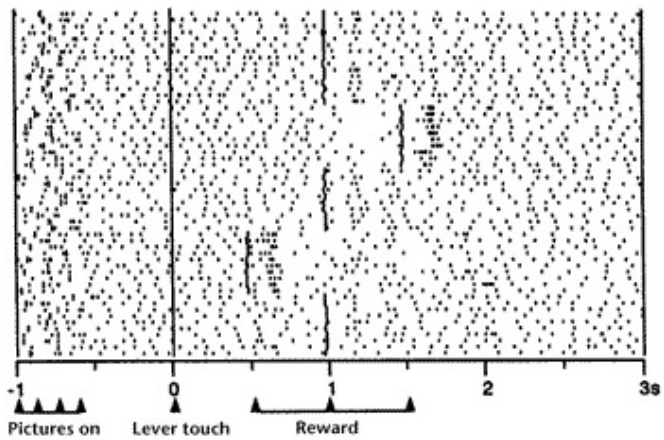
Belief states in the dopamine system

- Claim: belief states resurrect the viability of TD learning in the face of partial observability.

Belief states in the dopamine system

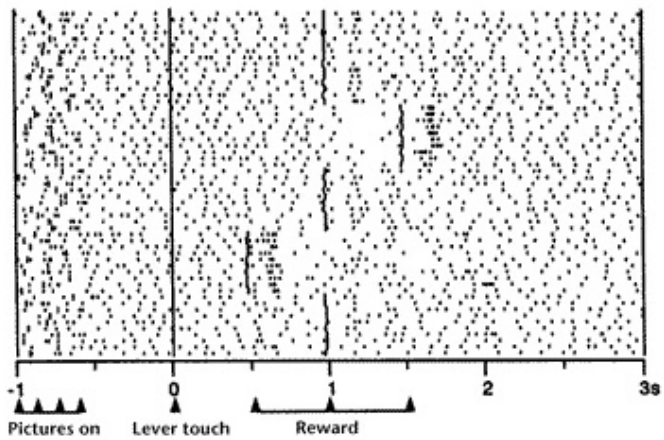
- Claim: belief states resurrect the viability of TD learning in the face of partial observability.
- If the brain uses this strategy, then we should expect to see signatures of belief states in phasic dopamine activity (putatively reporting the TD error).

Belief states in the dopamine system



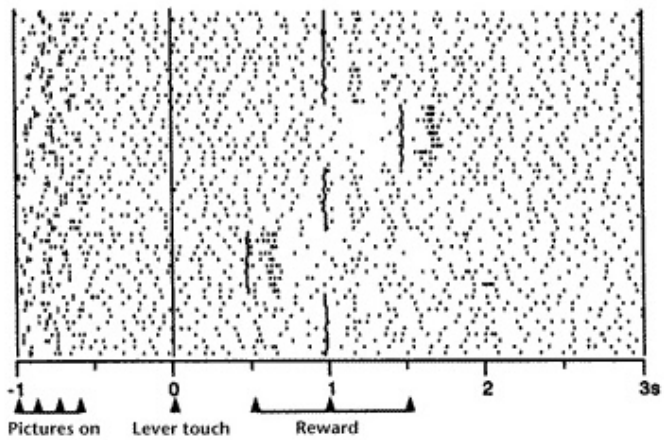
- Monkeys were trained with a regular ISI and then tested with irregular ISIs [Hollerman & Schultz 1998].

Belief states in the dopamine system



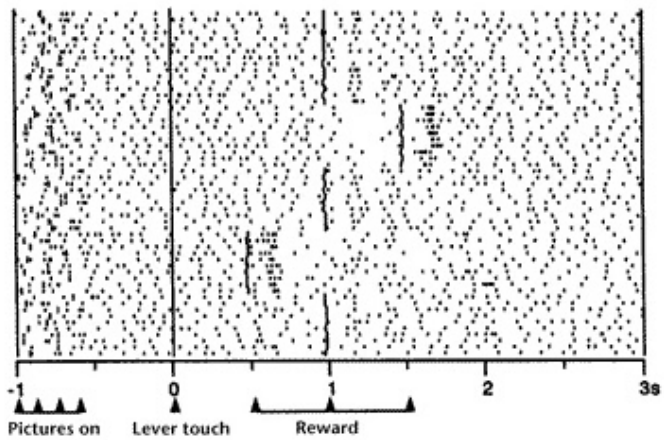
- Monkeys were trained with a regular ISI and then tested with irregular ISIs [Hollerman & Schultz 1998].
- Early reward deliveries produced a pronounced excitation at the unexpected time of reward, yet no subsequent dip at the expected time of reward.

Belief states in the dopamine system



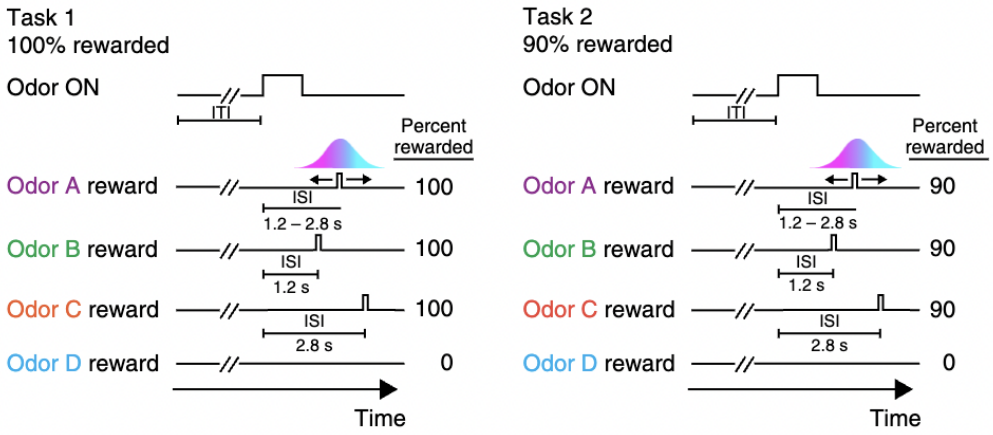
- Monkeys were trained with a regular ISI and then tested with irregular ISIs [Hollerman & Schultz 1998].
- Early reward deliveries produced a pronounced excitation at the unexpected time of reward, yet no subsequent dip at the expected time of reward.
- This is puzzling because it would seem that this should produce a negative prediction error.

Belief states in the dopamine system



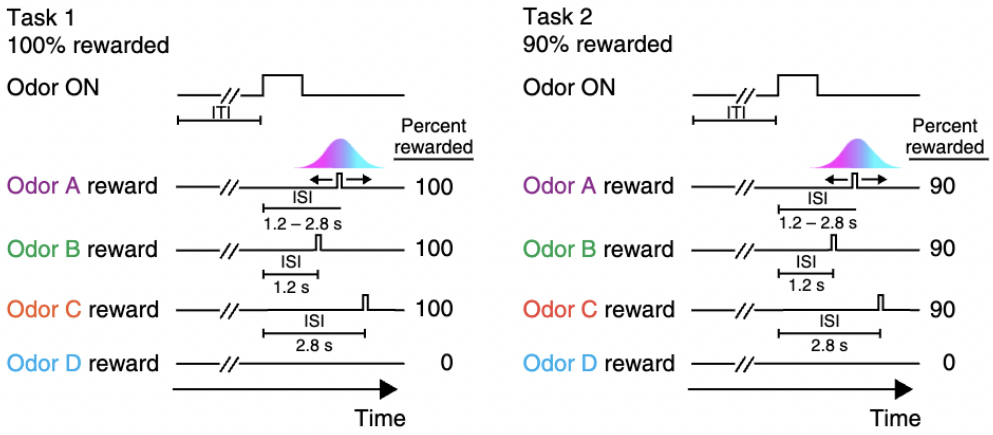
- Monkeys were trained with a regular ISI and then tested with irregular ISIs [Hollerman & Schultz 1998].
- Early reward deliveries produced a pronounced excitation at the unexpected time of reward, yet no subsequent dip at the expected time of reward.
- This is puzzling because it would seem that this should produce a negative prediction error.
- When the early reward occurs, the belief state should shift probability mass to the ITI state (no reward expected).

Belief states in the dopamine system



- Two tasks [Starkweather et al 2017], 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).

Belief states in the dopamine system



- Two tasks [Starkweather et al 2017], 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).
- Odor cue associated with a Gaussian-distributed delay.

Belief states in the dopamine system

- Fully observable Task 1: reward will always arrive, so the animal's reward expectation should *grow* as time elapses.

Belief states in the dopamine system

- Fully observable Task 1: reward will always arrive, so the animal's reward expectation should *grow* as time elapses.
- Corollary: reward prediction error should be smaller for later rewards.

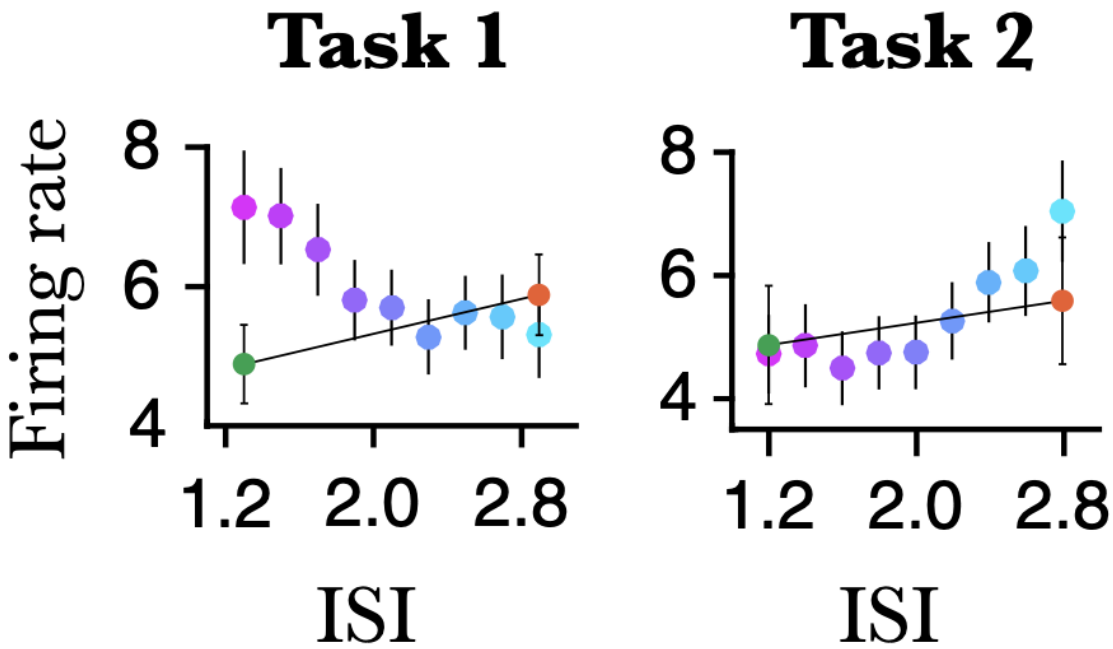
Belief states in the dopamine system

- Fully observable Task 1: reward will always arrive, so the animal's reward expectation should *grow* as time elapses.
- Corollary: reward prediction error should be smaller for later rewards.
- Partially observable Task 2: the reward may never arrive, so the animal's reward expectation should *shrink* as time elapses, due to the increasing probability that it's currently in an omission trial.

Belief states in the dopamine system

- Fully observable Task 1: reward will always arrive, so the animal's reward expectation should *grow* as time elapses.
- Corollary: reward prediction error should be smaller for later rewards.
- Partially observable Task 2: the reward may never arrive, so the animal's reward expectation should *shrink* as time elapses, due to the increasing probability that it's currently in an omission trial.
- Corollary: the prediction error should be larger for later rewards.

Response of dopamine neurons to reward after variable ISIs



[Starkweather et al 2017]

Emergent belief states

- 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.

Emergent belief states

- 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.
- Alternative approach: build an expressive function approximator that can be trained to compute representations suitable for reward prediction.

Emergent belief states

- 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.
- Alternative approach: build an expressive function approximator that can be trained to compute representations suitable for reward prediction.
- Because belief states are sufficient for reward prediction, this could plausibly yield belief-like representations.

Emergent belief states

- Value RNN: recurrent network trained for reward prediction [Hennig et al 2023].

Emergent belief states

- Value RNN: recurrent network trained for reward prediction [Hennig et al 2023].
- Learned representations are "belief-like" in the sense that their dynamics resemble the dynamics of belief state updating, and belief states can be approximately decoded from them.

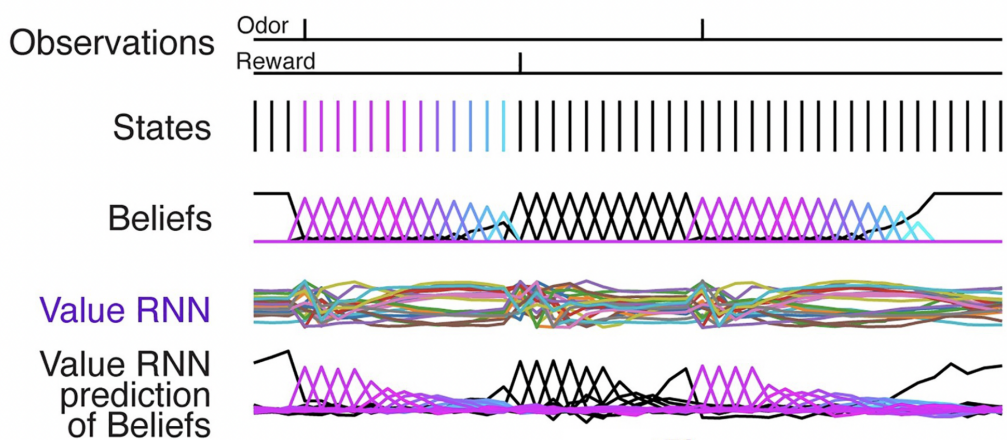
Emergent belief states

- Value RNN: recurrent network trained for reward prediction [Hennig et al 2023].
- Learned representations are "belief-like" in the sense that their dynamics resemble the dynamics of belief state updating, and belief states can be approximately decoded from them.
- Value RNN representations are lower dimensional than the belief states, retaining only the information in memory needed to predict reward.

Emergent belief states

- Value RNN: recurrent network trained for reward prediction [Hennig et al 2023].
- Learned representations are "belief-like" in the sense that their dynamics resemble the dynamics of belief state updating, and belief states can be approximately decoded from them.
- Value RNN representations are 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.

Belief state decoding from a recurrent neural network



[Hennig et al 2023]

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.

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.

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.

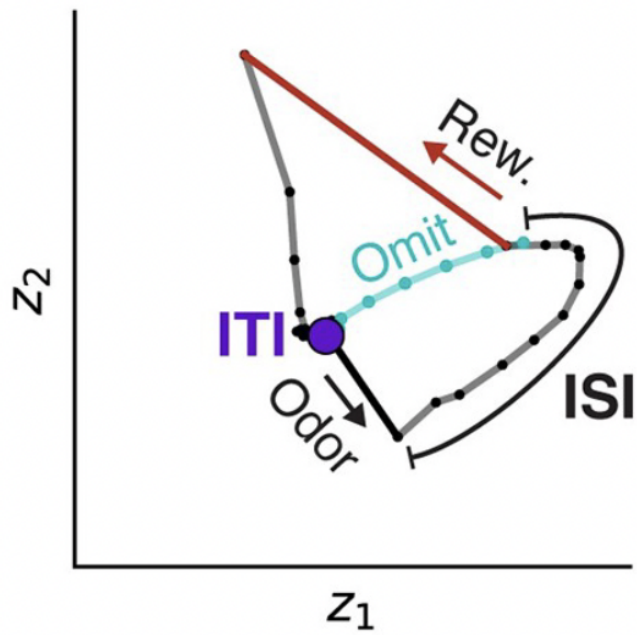
Recurrent neural network models of working memory

- Dynamical analysis of the Value RNN trained on the Pavlovian protocol reveals a single "fixed point" corresponding to the ITI, which is stable in the absence of sensory input.

Recurrent neural network models of working memory

- Dynamical analysis of the Value RNN trained on the Pavlovian protocol reveals a single "fixed point" corresponding to the ITI, which is stable in the absence of sensory input.
- When an odor is presented, the activity is 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).

Dynamics of the Value RNN



[Hennig et al 2023]

Recurrent neural network models of working memory

- Dynamics exhibit both memory maintenance (the trajectory encodes a trace of the stimulus) and time-keeping (the trajectory encodes a representation of elapsed time).

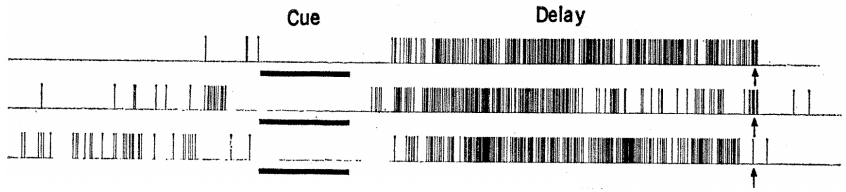
Recurrent neural network models of working memory

- 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.

Recurrent neural network models of working memory

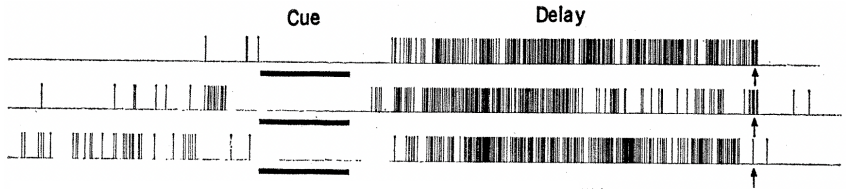
- Dynamics exhibit both memory maintenance (the trajectory encodes a trace of the stimulus) and time-keeping (the trajectory encodes a representation of elapsed time).
- This reflects the underlying state structure: the stimulus trace corresponds to the macrostate, and the temporal trace corresponds to the microstate.
- 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.

Recurrent neural network models of working memory



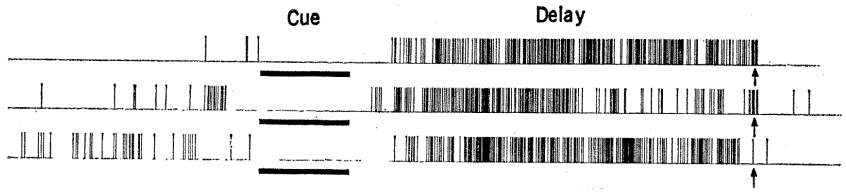
- Some prefrontal neurons exhibit persistent activity when animals need to store short-term memory (e.g., in a delayed matched-to-sample task).

Recurrent neural network models of working memory



- Some prefrontal neurons exhibit persistent activity when animals need to store short-term memory (e.g., in a delayed matched-to-sample task).
- Standard mechanism is recurrent excitation that produces "reverberating" activity, allowing stimulus representation to be maintained even after the stimulus has disappeared.

Recurrent neural network models of working memory



- Some prefrontal neurons exhibit persistent activity when animals need to store short-term memory (e.g., in a delayed matched-to-sample task).
- Standard mechanism 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 robust to small perturbations.

Beyond persistent activity

- Persistent activity is classically understood to mean single neurons that are activated by a stimulus and continue firing during a delay period.

Beyond persistent activity

- Persistent activity is classically understood to mean single neurons that are activated by a stimulus and continue firing during a delay period.
- However, 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.

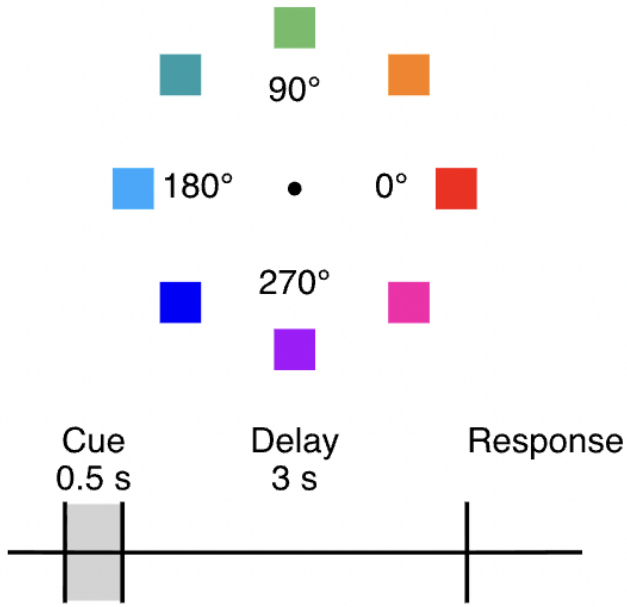
Beyond persistent activity

- Persistent activity is classically understood to mean single neurons that are activated by a stimulus and continue firing during a delay period.
- However, 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.
- 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.

Beyond persistent activity

- Persistent activity is classically understood to mean single neurons that are activated by a stimulus and continue firing during a delay period.
- However, 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.
- 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 does the prefrontal cortex stably maintain information in memory without widespread classical persistent activity?

Oculomotor delayed response task



[Constantinidis et al 2001; Murray et al 2017]

Subspace analysis

- Basic idea: decompose delay period activity into low-dimensional projections (principal components), averaging over some experimental factors but not others. These define different "subspaces" in the same neural population, carrying information about different factors.

Subspace analysis

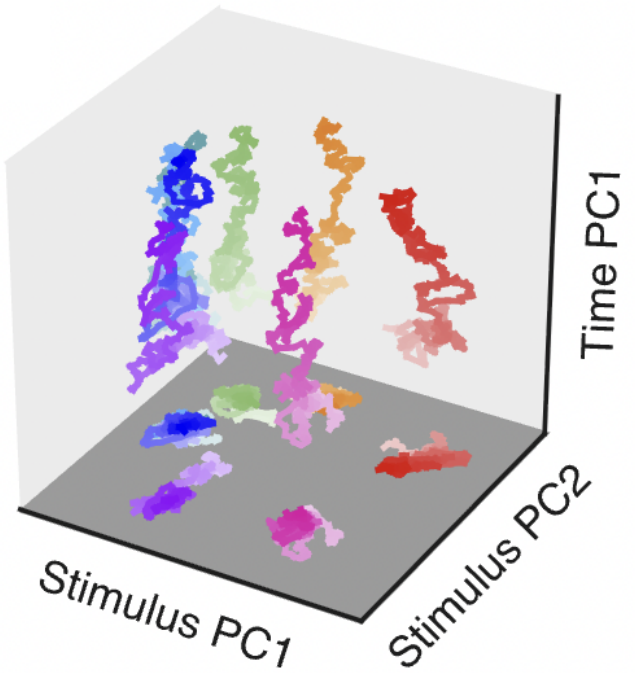
- Basic idea: decompose delay period activity into low-dimensional projections (principal components), averaging over some experimental factors but not others. These define different "subspaces" in the same neural population, carrying information about different factors.
- **Mnemonic subspace**: PCA decomposition of stimulus-by-neuron matrix, constructed by averaging delay activity for each stimulus-neuron pair; this discards time information but retains stimulus information.

Subspace analysis

- Basic idea: decompose delay period activity into low-dimensional projections (principal components), averaging over some experimental factors but not others. These define different "subspaces" in the same neural population, carrying information about different factors.
- **Mnemonic subspace**: PCA decomposition of stimulus-by-neuron matrix, constructed by averaging delay activity for each stimulus-neuron pair; this discards time information but retains stimulus information.
- **Dynamic subspace**: PCA decomposition of time-by-neuron matrix, constructed by averaging across stimuli for each time-neuron pair; this discards stimulus information but retains time information.

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



[Murray et al 2017]

Models

- Several standard models cannot capture these results.

Models

- Several standard models cannot capture these results.
- **Stable attractor model**: each neuron is tuned to a particular location. Neurons tuned to nearby locations excite one another, while inhibiting neurons tuned to distant locations.

Models

- Several standard models cannot capture these results.
- **Stable attractor model**: 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.

Models

- Several standard models cannot capture these results.
- **Stable attractor model**: 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.
- The model does not show much temporal variation during the delay period, in contrast to the prefrontal population.

Models

- **Stable subspace model**: similar to stable attractor model, but the connectivity structure is organized to have a dynamic subspace in addition to the mnemonic subspace.

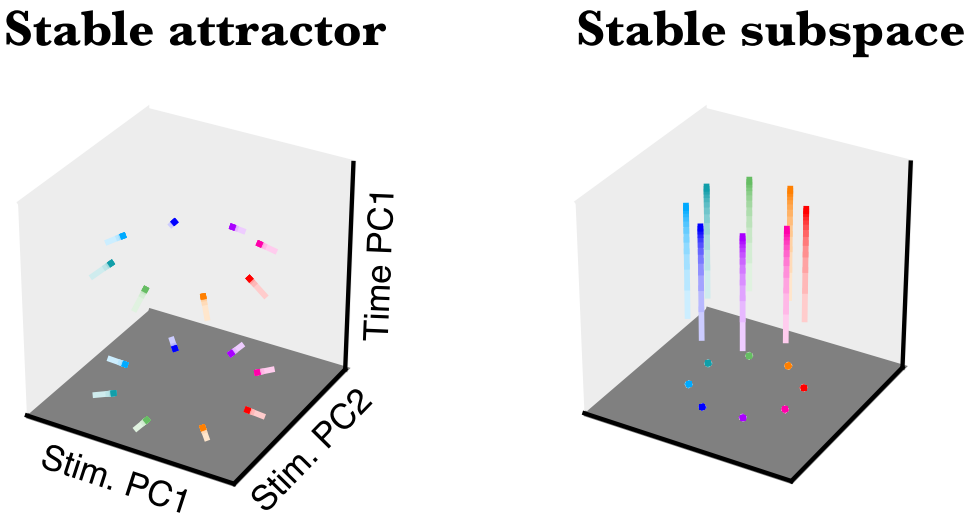
Models

- **Stable subspace model**: similar to stable attractor model, but the connectivity structure is organized to have a dynamic subspace in addition to the mnemonic subspace.
- Activity varies across time in the dynamic subspace, while remaining stable in the mnemonic subspace.

Models

- **Stable subspace model**: similar to stable attractor model, but the connectivity structure is organized to have a dynamic subspace in addition to the mnemonic subspace.
- Activity varies across time in the dynamic subspace, while remaining stable in the mnemonic subspace.
- This reproduces the geometry of the neural data when population activity is projected onto the principal components.

Subspace analysis of two models during the ODR task



[Murray et al 2017]

What are subspaces and why do we have them?

- From a decoding perspective, subspaces are just different choices of linear readout weights.

What are subspaces and why do we have them?

- From a decoding perspective, subspaces are just different choices of linear readout weights.
- The same network can support downstream processing of both stimulus and temporal information.

What are subspaces and why do we have them?

- From a decoding perspective, subspaces are just different choices of linear readout weights.
- 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.

What are subspaces and why do we have them?

- From a decoding perspective, subspaces are just different choices of linear readout weights.
- 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.
- Use of multiple subspaces 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.

Long-term memory

- The kinds of memory we've been talking about are short-lived, limited fundamentally by leakage and noise—processes that can only be resisted by refreshing of the activity state encoding information.

Long-term memory

- The kinds of memory we've been talking about 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.

Long-term memory

- The kinds of memory we've been talking about 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).

Long-term memory

- The kinds of memory we've been talking about 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).
- What kind of partial observability is being solved by a long-term memory system?

Long-term memory

- 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 $\{\tilde{t}_1, \ldots, \tilde{t}_N\}$.

Long-term memory

- 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 $s(t) = (x(\tilde{t}_1), \ldots, x(\tilde{t}_N))$.

Long-term memory

- 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 $s(t) = (x(\tilde{t}_1), \ldots, x(\tilde{t}_N))$.
- 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.

Long-term memory

- 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 $s(t) = (x(\tilde{t}_1), \ldots, x(\tilde{t}_N))$.
- 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.
- Long-term memory must be able to retrieve information arbitrarily far in the past.

Long-term memory

- Solution: single high-capacity database of past observations.

Long-term memory

- Solution: 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).

Long-term memory

- Solution: 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.

The key-value data structure

- Transformers use a parallelized “attention” mechanism to select past inputs for retrieval.

The key-value data structure

- 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.

The key-value data structure

- 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.
- Input mapped to a query vector $q(t)$ in the same address space as the keys, matched by a similarity function $S(k, q)$.

The key-value data structure

- 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.
- Input mapped to a query vector $q(t)$ in the same address space as the keys, matched by a similarity function $S(k, q)$.
- Similarity values are 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 and $\sigma[\cdot]$ is a "separator" function that can amplify strong matches and suppress weak matches.

The key-value data structure

- 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.
- Input mapped to a query vector $q(t)$ in the same address space as the keys, matched by a similarity function $S(k, q)$.
- Similarity values are 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 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.

Attention

- Attention score defines the retrieval strength for content (the value vector) linked to each past input.

Attention

- Attention score defines the retrieval strength for content (the value vector) linked to each past input.
- Retrieved value is a linear composite of past values weighted by their attention scores:

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

Attention

- Attention score defines the retrieval strength for content (the value vector) linked to each past input.
- Retrieved value is a linear composite of past values weighted by their attention scores:

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

- Because the attention vector is normalized, we can potentially interpret it as a belief state—i.e., posterior probabilities over a hidden state.

Attention

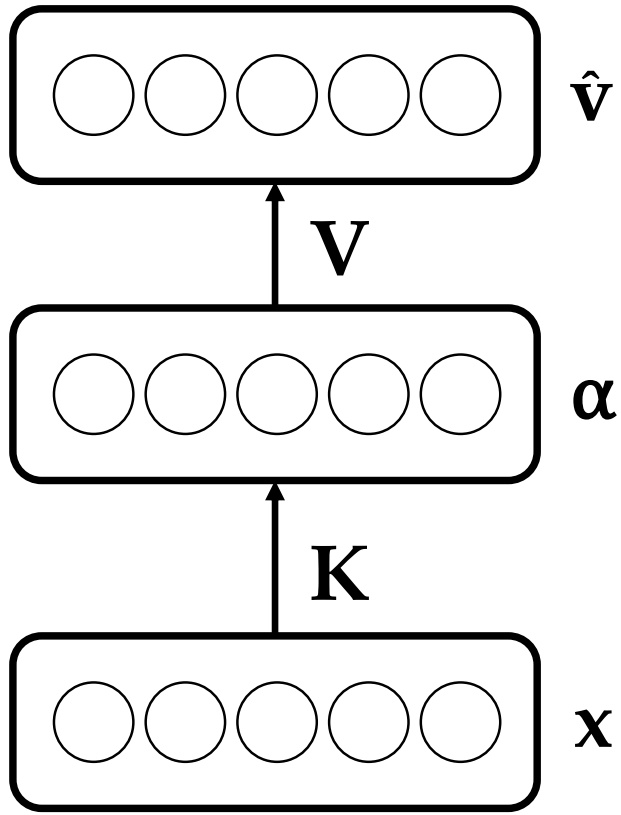
- Attention score defines the retrieval strength for content (the value vector) linked to each past input.
- Retrieved value is a linear composite of past values weighted by their attention scores:

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

- Because the attention vector is normalized, we can potentially interpret it as a belief state—i.e., posterior probabilities over a hidden state.
- Hidden state $s \in \{1, \ldots, t-1\}$ 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.

Neural architecture for key-value memory

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.



The key-value data structure

- 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.

The key-value data structure

- 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.
- Hippocampus does not itself encode memory content, but rather serves as an address space for similarity-based indexing.

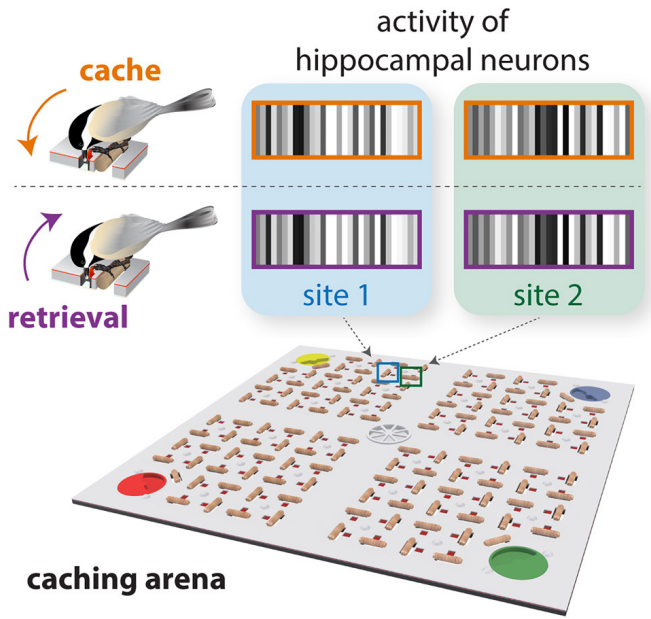
The key-value data structure

- 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.
- Hippocampus does not itself encode memory content, but rather serves as an address space for similarity-based indexing.
- Memory retrieval relies on reinstatement of activity in cortex, which depends on the hippocampus.

The key-value data structure

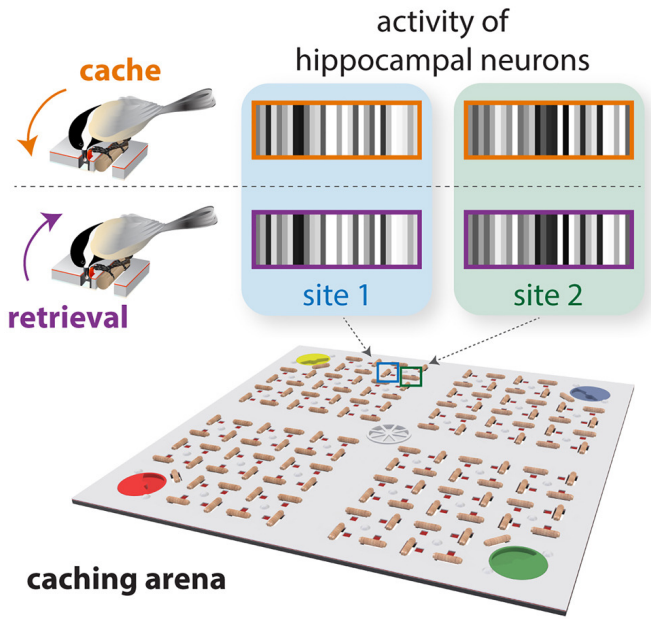
- This architecture can be mapped onto the hippocampal-cortical system, where the hippocampus encodes attention scores, which are used to retrieve content (value) stored in cortex.
- Hippocampus does not itself encode memory content, but rather serves as an address space for similarity-based indexing.
- Memory retrieval relies on reinstatement of activity in cortex, which depends on the hippocampus.
- 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.

How is the hippocampal address space organized?



- Food caching birds (chickadees) can remember hundreds of cache locations over several weeks with millimeter precision.

How is the hippocampal address space organized?



- Food caching birds (chickadees) can remember hundreds of cache locations over several weeks with millimeter precision.
- Hippocampal neurons exhibit a sparse and transient "barcode" pattern unique to individual caches and reinstated at retrieval [Chettih et al 2024].

How is the hippocampal address space organized?

- Barcodes can be captured by an RNN with random recurrent weights, transiently amplified during memory storage to produce an effectively random pattern of activity [Fang et al 2025].

How is the hippocampal address space organized?

- Barcodes can be captured by an RNN with random recurrent weights, transiently amplified during memory storage to produce an effectively random pattern of activity [Fang et al 2025].
- By associating these patterns with sensory input (the input-to-key mapping in the key-value framework), the patterns are reactivated when a cache location is approached.

Study question

How can we reconcile the different functional views of the hippocampus (predicting, modeling, remembering) that have been discussed in this chapter and earlier chapters?

Summary

- Memory is a solution to partial observability—the unavailability of state information, which violates the Markov property and cripples efficient computation.

Summary

- Memory is 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?

Summary

- Memory is 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.

Summary

- Memory is 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. **Memory is a quest for state**.