Neural Replay in Representation, Learning and Planning

Yunzhe Liu

Supervised by:

Professor Ray Dolan FRS
Professor Tim Behrens FRS

A thesis submitted in partial fulfilment for the degree of Doctor of Philosophy
At the Institute of Neurology, Faculty of Brain Sciences, University College London, UK

September 2020

天下无心外之物，如此花树在深山中自开自落，于我心亦何相关？

你未看此花时，此花与汝心同归于寂。你来看此花时，则此花颜色一时明白起来。

—— 王阳明

Q: If there is nothing under heaven external to the mind, these flowering trees on the high mountain blossom and drop their blossoms of themselves. What have they to do with my mind?

A: Before you look at these flowers, they and your mind are in the state of silent vacancy. As you come to look at them, their colors at once show up clearly. From this you can know that these flowers are not external to your mind.

—— *Wang Yangming*

Declaration

I, Yunzhe Liu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis. It has not been previously submitted, in part or whole, to any university or institution for any degree, diploma or other qualification.

By the University College London guidelines, this thesis does not exceed 100,000 words, and it contains less than 150 figures.

Signature:

Date:

Yunzhe Liu

London, UK

17th September 2020

Abstract

Spontaneous neural activity is rarely the subject of investigation in cognitive neuroscience. This may be due to a dominant metaphor of cognition as the information processing unit, whereas internally generated thoughts are often considered as noise. Adopting a reinforcement learning (RL) framework, I consider cognition in terms of an agent trying to attain its internal goals. This framework motivated me to address in my thesis the role of spontaneous neural activity in human cognition.

First, I developed a general method, called temporal delayed linear modelling (TDLM), to enable me to analyse this spontaneous activity. TDLM can be thought of as a domain general sequence detection method. It combines nonlinear classification and linear temporal modelling. This enables testing for statistical regularities in sequences of neural representations of a decoded state space. Although developed for use with human non-invasive neuroimaging data, the method can be extended to analyse rodent electrophysiological recordings.

Next, I applied TDLM to study spontaneous neural activity during rest in humans. As in rodents, I found that spontaneously generated neural events tended to occur in structured sequences. These sequences are accelerated in time compared to those that related to actual experience (30 -50 ms state-to-state time lag). These sequences, termed replay, reverse their direction after reward receipt. Notably, this human replay is not a recapitulation of prior experience, but follows sequence implied by a learnt abstract structural knowledge, suggesting a factorized representation of structure and sensory information.

Finally, I test the role of neural replay in model-based learning and planning in humans. Following reward receipt, I found significant backward replay of non-local experience with a 160 ms lag. This replay prioritises and facilitates the learning of action values. In a separate sequential planning task, I show these neural sequences go forward in direction, depicting the trajectory subjects about to take.

The research presented in this thesis reveals a rich role of spontaneous neural activity in supporting internal computations that underpin planning and inference in human cognition.

Impact Statement

The line of research presented in this thesis, starts with introduction of a general sequence detection method, followed by its contribution to an understanding of structured spontaneous neural activity, termed replay, in mental representation, model-based learning and planning. The thesis ends with a call for new directions in cognitive neuroscience.

Having a method (TDLM) to measure fast neural sequences in non-spatial spaces in humans non-invasively is of great importance. This is because an ability to detect replay outside of contexts related to spatial processing broadens its potential scope. This includes enabling experiments that investigate broader areas of cognitive neuroscience, especially measurement of human replay in contexts that are not possible in rodents (such as language, numerical cognition, or flexible executive tasks). The method is also domain general, although developed in the context of human neuroimaging, it can be applied to other data sources, including rodent electrophysiology recordings. It has the potential to facilitate novel cross-species investigations.

Next, I show that human replay facilitates building a factorized representation. In other words, I demonstrate that multiple representations of different aspects of events are replayed simultaneously, and these basic representations can be recombined to make new events. This is important because factorised representations provide a powerful means for generalising knowledge. With factorised representations, individual experiences can be decomposed into parts and these parts can be meaningfully recombined into a vast number of ways – a form of combinatorial generalisation which has the potential to dramatically improve learning and inference.

Finally, I show human replay facilitates non-local learning and supports model-based planning. I show that replay exploits knowledge of the world (task structure) to perform non-local mental simulations that support model-based computations. This suggests human cognition is a not a simple matter of processing external stimuli but instead uses rich internal computations to support complex adaptive behavior.

I conclude by proposing a new framework and experimental paradigm that takes its inspiration from model-based reinforcement learning, as well as a call for a more unified approach within cognitive neuroscience research. I suggest a unification of terminology. coupled with a new metaphor in human psychology (e.g., a RL agent), can open doors to a renewed research focus on the functional role of the brain's internal computations.

Acknowledgements

It is hard to put an end mark without being emotional. This has been an amazing 4 years of adventure, one that I still find surreal today.

I remember the first conversation with Tim. He said, “no one is smarter than anyone else, the only difference is knowledge”. This sets up the tone of my whole PhD - not to be afraid of the unknown, and not to feel incompetent when facing the challenge.

The same goes with Ray. I asked what I should do as a PhD student when I first came here, Ray said “this is your PhD, spend the first three months do nothing but think, if you cannot find it, I will help you”. And indeed, Ray’s help has sustained throughout my PhD, from helping me finding an accommodation, to introducing me to the richness of the UK culture, and the Western at large, while teaching me English.

The same gratitude goes to Zeb. I still remembered the nerve-wracking feeling when the first time I tried to explain my thoughts to, “the smartest guy ever”. The idea is of course naïve, and the expression is nothing logic, but a stream of consciousness. Yet, Zeb got it, and soon turn it into a great project that define my PhD.

There are too many great minds at FIL, MPC and FMRIB that I cannot count. To name a few. To Laurence, for always being kind and supportive, and took me to the first-time-ever brain meeting dinner, despite me being awkwardly quiet, and only knew to order fishes and chips. To Elliott, for being a great friend and colleague, and took a leap of faith in the method we have developed and managed a home run. To Matt, for always being there to chat, both in science and life at large. And to Toby, Evan, Rani, Jessica, and many more, it is a great pleasure to work with you all. I am eternally grateful to your trust and companion.

The last thank goes to my family, especially to my wife, Wanjun. My life will be miserable without you. Thank you for always being there for me, even it means to go abroad for 5 years. Also, to my father and mother. I know you are still upset that I did not choose finance, but hopefully, I can prove my use in 10 years’ time, although not by money I am afraid.

Publications from PhD work

(* co-first author)

1. **Liu, Y.,** Mattar, M., Behrens, T. E., Daw, N., Dolan, R.J. (2020) Experience replay supports nonlocal learning. ***bioRxiv*** *(in revision)*

2. **Liu, Y.,** Dolan, R. J., Kurth-Nelson, Z., Behrens, T. E. (2020) Temporal delayed linear modelling (TDLM): Measuring replays in both rodents and humans. ***bioRxiv*** *(in revision)*

3. Nour, M.*, **Liu, Y.*,** Arumuham, A., Kurth-Nelson, Z., Dolan, R. (2020) Impaired neural replay of inferred relational structure in schizophrenia. (*under review)*

4. Higgins, C.*, **Liu, Y.*,** Vidaurre, D, Kurth-Nelson, Z., Dolan, R. J., Behrens, T. E., Woolrich, M (2020) Replay bursts coincide with activation of the default mode and parietal alpha network. ***bioRxiv (Neuron****, accepted)*

5. Wimmer, G. E.*, **Liu, Y.*,** Vehar, N., Behrens, T. E., Dolan, R. J. (2020) Episodic memory retrieval success is associated with rapid replay of episode content. ***Nature Neuroscience***, 1-9.

6. **Liu, Y.,** Dolan, R. J., Kurth-Nelson, Z., Behrens, T. E. (2019) Human replay spontaneously reorganises experience. ***Cell,*** 178(3), 640-652.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 The general question

In the era of modern cognitivism, we study cognition within an information processing metaphor, where the core notion is that the mind processes an informational input from the outside world and generates action commands thereafter [1]. This is, arguably, consistent with a stimulus-response framework that can be traced back to Behaviourism [2]. In the modern time, while cognition is no longer considered solely in these terms, there is still a dominant metaphor in thinking cognition as a responsive process, triggered by external stimuli.

In neuroscience, we rarely consider spontaneous neural activity as reflecting cognition, particularly under a stimulus-response framework. In fact, in another area of neuroscience, spontaneous neural activity is referred to as "default mode", based on observations that some expressions of neural activity in specific brain regions are deactivated by task demands, as compared to rest [3].

In psychology, there is a line of research inspecting the internal mental states, especially after the 'cognitive revolution' [4]. For example, Burke, et al. [5] studied free recall in neurosurgical patients and found both intracranial theta and high-frequency activity in the temporal lobe are associated with spontaneous episodic retrieval. Such line of research on mental state, does move away from stimuli evoked response, but it is still task related, it is not during rest.

During rest, we often have random thoughts that are not in any way relevant to any current task or goal. But are random thoughts, or spontaneous neural activity useful to the task, e.g., supporting behavior? To answer this, we need a different metaphor to how we think about the brain and cognition.

1.2 Reinforcement learning

Rather think of the brain as a passive computation machine that is responsive to external input, we need to think of it as more like an agent driven by internal goals. Ideally, this goal should be general. For this, we turn to the framework of reinforcement learning (RL) [6]. RL describes how an agent should behave in order to maximize long-term, cumulative reward.

To answer whether spontaneous neural activity is instrumental to the task goal. I begin by first describing RL, and then formulate this question in RL terms. Here, I appreciate that RL is also by itself limited. Nevertheless, it benefits from providing a common language that can be expressed in rigorous mathematical terms [7].

The concept of RL can differ across fields of enquiry. In machine learning, it is one of three dominate paradigms, together with supervised and unsupervised learning. In control theory, RL can be seen as a particular solution to model-based planning, with less of a focus on the learning or the model-free part of RL. Within psychology, researchers often see RL as a synonym for Rescorla–Wagner [8], a model of classical conditioning (which, in my opinion, is a disservice to both RL and Rescorla -Wagner [9]). I will use RL in its broad sense and borrow its language, concepts and general framework, without committing to the specificities of certain algorithms [6].

The general problem in RL is usually characterized by a state space, S, where the agent has a set of actions, A, that they can choose from. Normally the goal is to learn how to choose specific action a, at individual state, s, to maximize the state-action value, $Q(s,a)$. $Q(s,a)$ comprises both the immediate reward received conditional on the current action, and the expected (discounted) sum of future reward. The latter part is the crucial difference between RL and typical associative learning model, e.g., Rescorla–Wagner. It also sets up a requirement for cognition not just on current information input (e.g., current state, action or reward), but have to consider information non-locally.

The state space characterizes the relationship between states, $T = p(s'|s,a)$, The probability of transitioning to next state s' is determined by, and only by current state and action, s, a, a property referred to as Markovian. This is the only hard requirements on state space. It is possible to build difference state spaces for the same task. It turns out how to select the right task representation, characterized by T, is crucial for the efficiency of learning and generalization in novel contexts [10].

Policy, π, describes the probability of choosing each action in each given state, $\pi = p(a|s)$. If we denote the immediate reward upon on the current action, as r, and the discounting factor on future reward as γ. Then we can describe the most desirable policy in given state $\pi^*(s)$:

$$\pi^*(s) = \underset{a \epsilon A}{\operatorname{argmax}}\, Q^*(s,a) \tag{1}$$

$$\text{Where, } Q^*(s,a) = r + \gamma \textstyle\sum_{s' \in S} p(s'|s,a) \max_{a'} Q^*(s',a') \tag{2}$$

This equation describes how the best action can be taken with full knowledge of the state space, i.e., T, and the state-action value, $Q(s,a)$. In reality, however, $Q(s,a)$ is not given and can change due to the stochasticity of reward. In an ever-changing environment, this creates a constant task – updating the estimate of $Q(s,a)$. This can be done based on the difference between feedback and current estimate. For example,

$$Q\ (s,a) \leftarrow Q\ (s,a) + \alpha\ [r + \gamma \max_{a'} Q(s',a') - Q\ (s,a)] \tag{3}$$

This is called Q learning [11], where α is learning rate, determines how much the agent should learn from this discrepancy. This learning method is proven to converge on the optimal $Q^*(s,a)$ [11]. Notably, the outcome, and action are normally separated both in space and time. For example, a bad action in the early playing of GO [12], might lead to final failure several hours and many individual plays later. How to update the $Q\ (s,a)$ in early states based on a final outcome is a problem called credit assignment.

In situations where the $Q^*(s,a)$ is unknown, how to select the best action is also an active field of research. It is dangerous to simply take actions that $\underset{a\epsilon A}{\operatorname{argmax}} Q^*(s,a)$ based on current estimate (i.e., greedy policy), because the estimate of $Q^*(s,a)$ depends on the $Q^*(s',a')$, i.e., the future state-action value. Being greedy based on current estimate will likely lead to suboptimal behavior given future reward might be higher if we choose a different action in current state (albeit it might lead to less immediate reward). This is a problem known as sequential decision making, i.e., where the optimal decision at current state also depends on decisions at future states.

Both the problem of credit assignment in learning, and sequential decision-making in planning, are hard to solve if we only consider information input from the current state, action or reward. Also, RL operates on a mental representation of task space. This need not to be exactly the same as the visual input itself. In fact, how to represent a task state plays crucial role in generalization and inference in novel context. For example, we know where to go when we land at a new airport, this is because we don't consider the particular colour or texture of the wall important, but instead believe the general layout and function of all airports have similarities. Thus, we know that for any airport we will have to first go through boarder control before we can claim our baggage. We can infer this because we carry a general representation of the layout of the airport. A, perhaps more fundamental, question remains un-answered: how we build efficient task representations.

It is unknown how our brains solve all these problems. One thing abundantly clear is that none of this can be accomplished if we don't allow our cognition to detach from processing current information.

1.3 Replay

An intriguing and remarkable neural phenomenon found in rodent is that cells in hippocampus, those that normally represent given locations in space, also fire spontaneously and sequentially during rest [13-15]. This firing pattern is found not to be random but recapitulates past or future trajectories, termed replay [13,16-19]. Replay is interesting because it is an expression of structured neural activity that is detached from current input. In rodents, both the direction and content of replay are modulated by reward and task demands [20,21].

In neuroscience, early study of replay focus on its time compression feature[14,15], such feature make it suitable to support Hebbian learning, e.g., by reactivating memory in a short time window, so that spike timing dependent plasticity can help form new memories [22]. Later studies have found a broader role of replay that goes beyond memory consolidation [16], For example, Ambrose, et al. [20] found replay, especially reverse replay during outcome receipt supports value learning. Ólafsdóttir, et al. [23] found hippocampal place cells construct reward related sequences through unexplored space, potentially supporting planning or mental simulation. Interestingly, neural replay is not constrained to hippocampus. Similar phenomena have also been found in entorhinal cortex[24], and visual cortex[25]. In situations, those replays coordinate with hippocampal replay[26,27].

While replay studies are mostly conducted in rodents. There are also hint that such replay might also exist in humans. For example, using intracranial electroencephalography in humans, Vaz, et al. [28] found replay of spiking sequences in the temporal lobe during associative memory retrieval (although it is not time compressed). Using non-invasive neuroimaging, e.g., functional magnetic resonance imaging (fMRI), Tambini, et al. [29] showed enhanced brain correlations during rest are related to memory for recent experiences, and in a subsequent study, using transcranial magnetic stimulation, they demonstrated that the functional connectivity between hippocampus and lateral occipital cortex is causally related to memory consolidation [30]. Taking a step further, Schuck and Niv [31] have provided evidence of sequential reactivation of task states in human hippocampus during rest, although it is not clear what is the speed of such replay given the temporal resolution constraint of fMRI.

The replay like phenomenon is also implicated in RL theory. In one particular family of algorithms, called DYNA[32,33], all past experiences are proposed to be stored in a memory pool, or built into a model. During an updating phase, samples are drawn from this memory pool to facilitate learning of $Q(s,a)$ nonlocally, an idea that has empirically improved the performance of deep neural networks (e.g., AlphaGo, which beat the world Champions in GO [12]). A recent RL model suggests prioritizing memory access to this memory pool explain many replay related phenomena in rodents, where the prioritization is based on utility: namely, how much extra reward can be earned due to better choices [34].

What makes replay more interesting, perhaps, is that it might also help build efficient representations of a task. This echoes a notion that goes back to the idea of "cognitive map" from Tolman [35]. A cognitive map or world model is the representation of task space, it is about the *relational structure* representing the way different elements are connected to each other [10]. There are two important quantifies – states and their relationship. The state itself can change based on the goal, for example, representing airport as one single state is more convenient if you want to describe how to go from airport to hotel. On the other hand, we might want to represent it differently, if our goal is to figure out where to go to claim the luggage. Different state definitions therefore entail different relational structures, i.e., different T.

In rodents, replay has been shown to not be a simple function of experience, it can represent shortcuts that the rodent has not traversed [36]. This raises an interesting question, whether replay re-organizes experience in order to build suitable task representations based on different goals.

It is intriguing to think that this replay is a neural correlate of random thought, given it is spontaneous neural activity often seen during rest, as well as an expression of a sequential reactivation of neural representations of spaces. At neural level, seminal work in rodents and humans has shown that cognitive maps (or states) are encoded in the hippocampal-entorhinal system [26,27,37-41], regions where replay also occurs. If we can show similar replay phenomenon exist for non-spatial spaces in humans, it might suggest replay is a more generic footprint of cognition.

To answer whether random thoughts, or spontaneous patterns of structured neural activity are useful, I study the role of human replay in the framework of RL. First, I develop a method to measure sequences on any graphs where they are not necessarily reflecting physical space. The more long-term aim of this is to study arbitrary transitions in non-spatial tasks, a step in opening doors on to the investigation of higher cognition. This method needs to enable sequence measurement on non-invasive human neuroimaging data, and should also be applicable to invasive electrophysiology recording, commonly seen in rodent research, to facilitate cross-species comparisons.

Note, there are conceptual difference between replay and theta sequence in rodent literature [42-44], with replay mostly during rest, while theta sequences are mostly described during active navigation. In active spatial navigation, there is a continuous theta (4-10 Hz) background neural activity in rodent hippocampus [45]. Interestingly, the spiking of place cell relative to the background theta phase moves earlier when traverse the place field, a phenomenon called theta phase precession [46]. Also, unlike replay during rest, which can go in either forward or backward direction, and encode distant (non-local) information, Theta sequence is almost always forward, and encoding local information, sometimes termed "look ahead" signal [42]. Theta sequence is found to support spatial planning in rodents. Similar theta related coding is also found in humans. For example, Heusser, et al. [47] found a theta-gamma phase code coupling that support episodic sequence memory formation in humans, and Kahana, et al. [48] found theta oscillations in humans exhibit task dependence during virtual maze navigation. Human theta is in a range of 3-7 Hz and it is not continuous compared to rodents [49].

In human studies reported in this thesis there is no active movement involved, though some of the studies do look for sequence at decision time (i.e., not rest). For consistency and simplicity, I will call any sequential pattern of spontaneous neural activity found in humans – replay (during rest or planning/decision time). I will discuss implications of the presence of human sequences with respect to rodent findings in the end.

2 MEASURING SEQUENCES OF REPRESENTATIONS

2.1 Introduction

Human neuroscience has made remarkable progress in detailing the relationship between the representations of different stimuli during task performance [50,51]. At the same time, it is increasingly clear that at rest, off-task, brain activity is structurally rich and characterising this is important for understanding the neural underpinnings of cognition [52]. However, unlike the case for task-based activity, little attention has been given to techniques that can measure representational content or structure in this resting activity. Here, we introduce TDLM (temporal delayed linear modelling) as an analysis framework, based on linear modelling, that can characterize temporal structure of internally generated neural representations.

TDLM enables a detailed examination of sequential patterns in neural reactivation that are not tied to task events. This approach is inspired by evidence from the rodent literature of rich temporal structure in representational content of offline brain activity. A seminal finding in rodent electrophysiological research is "hippocampal replay" [13,14,17]. During rest and quiet wakefulness, place cells in the hippocampus (that signal self-location during periods activity) spontaneously recapitulate old, and explore new, trajectories through an environment [13,17]. These internally generated sequences are hypothesized to reflect a fundamental feature of neural computation across tasks [10,20,22,42].

Applying TDLM on non-invasive neuroimaging data, we, and others, have shown it is possible to measure spontaneous sequences of neural representations during rest in humans [53,54]. The results resemble key characteristic of rodent hippocampal replay and can inform key computational principles of human cognition [54].

In the following, we introduce the logic and mechanics of TDLM in detail. We first compare performance of alternative algorithms on synthetic data, where the ground truth is known (see methods on "simulating MEG data"). Subsequently, we apply the method to real neural data, both human magnetoencephalography (MEG) [53,54] and rodent hippocampal electrophysiological (ephys) recordings (see methods on "Human MEG dataset" and "Rodent hippocampal ephys for detecting theta sequence"). In relation to the latter, we show TDLM successfully reproduces key findings, including the presence of theta sequences [55]. We have also shown the applicability of TDLM to human EEG (Appendix 1).

TDLM is a general, and flexible, tool for measuring neural sequences. It facilitates cross-species investigations by linking large-scale measurements in humans to cellular measurements in non-human species. I will outline its potential for revealing abstract cognitive processes that extend beyond sensory representation, open possible new avenues of research in cognitive science. All code and facilities will be available at https://github.com/yunzheliu/TDLM.

2.2 Results

2.2.1 TDLM

2.2.1.1 Overview of TDLM

My primary goal is to test for temporal structure of neural representation in humans. I also want a method to measure sequence in other species (e.g., rodents) to facilitate cross-species investigation. This sequence detection method, therefore, needs to be domain general. I choose to measure sequences in a decoded state space (e.g., posterior estimated locations in rodents[56] or time course of object reactivations in humans[57]). This approach renders results from different data types more comparable.

A general sequence detection method, ideally, should (1) uncover structural regularity in the reactivation of neural activity, (2) control for confounds that are not of interest, or tied to certain data types, and (3) test whether this regularity conforms to an underlying hypothesized structure. To achieve these, I developed the method under the General linear modelling (GLM) framework, and call it temporal delayed linear modelling, i.e., TDLM.

TDLM works on a decoded state space but it still needs to take care of confounds inherent in the data from where the state space is decoded. This is one of the main focus of TDLM.

The starting point of TDLM is a set of n time series, each corresponding to a decoded neural representation of a variable of interest. This is what we call state space, X, with dimension of time by states. These time series could themselves be obtained in several ways, described in detail in a later section ("Getting the states"). The aim of TDLM is to identify task or structure related regularities in sequences of these representations.

Consider, for example, a task in which participants have been trained such that n=4 distinct sensory objects (A, B, C, and D) belong to a consistent order: $A \rightarrow B \rightarrow C \rightarrow D$ (Figure 2.1a, b). If we are interested in replay of this sequence during subsequent resting periods (Figure 2.1c, d), we might want to ask statistical questions of the following form: "Does the existence of a neural representation of A, at time T, predict the occurrence of a representation of B at time T+Δt", and similarly for $B \rightarrow C$ and $C \rightarrow D$.

In TDLM we ask such questions using a two-step process. First, for each of the n^2 possible pairs of variables X_i and X_j, we find the correlation between the X_i time series and the Δt-shifted X_j time series.

These n^2 correlations comprise an empirical transition matrix, describing how likely each variable is to be succeeded at a lag of Δt by each other variable (Figure 2.1e). Second, we correlate this empirical transition matrix with a task-related transition matrix of interest (Figure 2.1f). This produces a single number that characterizes the extent to which the neural data follow the transition matrix of interest, which we call 'sequenceness'. Finally, we repeat this entire process for all Δt of interest, yielding a measure of sequenceness at each possible lag between variables, and submit for statistical inference (Figure 2.1g).

Note that, for now, this approach decomposes a sequence (such as $A \rightarrow B \rightarrow C \rightarrow D$) into its constituent transitions and adds the evidence for each transition. It therefore does not require that the transitions themselves are sequential: $A \rightarrow B$ and $B \rightarrow C$ could occur at unrelated times, so long as the within-pair time lag was the same. In section "Multi-step sequences", we address how to strengthen the inference by looking explicitly for longer sequences.

2.2.1.2 Constructing the empirical transition matrix

In order to find evidence for state-to-state transitions at some time lag Δt, we could regress a time-lagged copy of one state, X_j, onto another, X_i:

$$X_j(t + \Delta t) = X_i(t)\beta_{ij} \tag{1}$$

Instead, TDLM includes all states in the same regression model for important reasons, detailed in section "Controlling confounds and maximizing sensitivity in sequence detection":

$$X_j(t + \Delta t) = \sum_{k=1}^{n} X_k(t)\beta_{kj} \tag{2}$$

In this equation, the values of all states X_k at time *t* are used in a single multilinear model to predict the value of the single state X_j at time $t + \Delta t$.

The regression described in Equation 2 is performed once for each X_j, and these equations can be arranged in matrix form as follows:

$$X(\Delta t) = X\beta \tag{3}$$

Each row of *X* is a timepoint, and each of the *n* columns is a state. $X(\Delta t)$ is the same matrix as *X*, but with the rows shifted forwards in time by Δt. β is an $n \times n$ matrix of weights – which we call the *empirical transition matrix*. β_{ij} is an estimate of the influence of $X_i(t)$ on $X_j(t + \Delta t)$, over and above variance that can be explained by other states at the same time.

To obtain β, we invert Equation 3 by ordinary least squares regression.

$$\beta = (X^T X)^{-1} X^T X(\Delta t) \tag{4}$$

This inversion can be repeated for each possible time lag ($\Delta t = 1, 2, 3, \ldots$), resulting in a separate empirical transition matrix β at every time lag. We call this step the first level sequence analysis.

2.2.1.3 Testing the hypothesized transitions

The first level sequence analysis assesses evidence for all possible state-to-state transitions. The next step in TDLM is to test for the strength of a particular hypothesized sequence, specified as a transition matrix, T_F. We therefore construct another GLM which relates T_F to the empirical transition matrix β. We call this step the second level sequence analysis:

$$\beta = \sum_{r=1}^{r} Z(r) * T_r \tag{5}$$

β is the empirical transition matrix, with dimension n by n, where n is the number of states. Each entry in β reflects the unique contribution of state i to state j at given time lag. r is the number of regressors. T_r is the regressor in the design matrix, each of which is a transition matrix, e.g., T_{auto}, T_{const}, T_F and T_B.

T_F and T_B are the transpose of each other (e.g., red and blue entries in Figure 2.1b), indicating transitions of interest in forward and backward direction, respectively. In physical space T_F and T_B would be the shifted diagonal matrices with ones on the first upper and lower off diagonals. T_{const} is a constant matrix that models away the average of all transitions, ensuring that any weight on T_F and T_B. $T_{auto}$is the identity matrix, T_{auto} models self-transitions to control for auto-correlation (equivalently, we could simply omit the diagonal elements from the regression).

Z is the weights of the second level regression, which is a vector with dimension of r by 1. Each entry in Z reflects the evidence strength of the hypothesized transitions in the empirical ones, i.e., sequenceness. Repeating the regression of Equation 5 at each time lag ($\Delta t = 1, 2, 3, \ldots$) results in time courses of the sequenceness as a function of time lag (e.g., the solid black line in Figure 2.1f), in which Z_F, Z_B are the forward and backward sequenceness respectively (e.g., red and blue lines in Figure 2.1g).

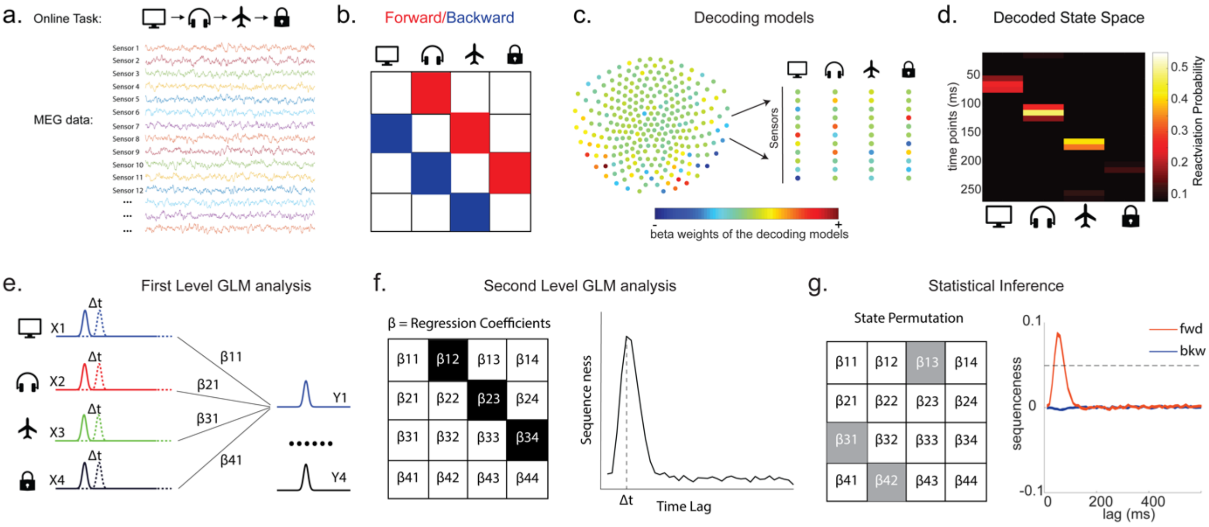
In many cases, Z_F and Z_B will be the final outputs of a TDLM analysis. However, it may sometimes also be useful to consider the quantity:

$$D = Z_F - Z_B \tag{6}$$

D contrasts forward and backward sequences to give a measure that is positive if sequences occur mainly in a forward direction, and negative if sequences occur mainly in a backward direction. This may be advantageous if, for example, Z_F and Z_B are correlated across subjects (due to factors such as subject engagement and measurement sensitivity). In this case, D may have lower cross-subject variance than either Z_F or Z_B, as the subtraction removes common variance.

Finally, to test for statistical significance, TDLM relies on a nonparametric permutation-based method. The null distribution is constructed by randomly shuffling the identities of the *n* states and re-calculating the second level analysis for each shuffle (Figure 2.1g). This approach allows us to reject the null hypothesis that there is no relationship between the empirical transition matrix and the task-defined transition of interest. Note that there are many wrong ways to perform permutations, which permute factors that are not exchangeable under the null hypothesis and will therefore lead to false positives. We will examine some of these later with simulations and real data. In some cases, it may be desirable to test slightly different hypotheses by using a different set of permutations; this will also be discussed later.

If the time lag Δt at which neural sequences exist is not known *a priori*, then it is critical to correct for multiple comparisons over all tested lags. This can be achieved by using the maximum Z_F across all tested lags as the test statistic. If we choose this test statistic, then any values of Z_F exceeding the 95th percentile of the null distribution can be treated as significant at $\alpha = 0.05$ (e.g., the grey dotted lined in Figure 2.1g).



β11	β12	β13	β14
β21	β22	β23	β24
β31	β32	β33	β34
β41	β42	β43	β44

β11	β12	β13	β14
β21	β22	β23	β24
β31	β32	β33	β34
β41	β42	β43	β44

Figure 2.1 Task design and illustration of TDLM

a, Task design in both simulation and real MEG data. Assuming there is one sequence, A->B->C->D, indicated by the four objects. During task, participants are shown the objects, and are asked to work out where the object fits in underling sequence while undergoing MEG scanning. A snapshot of MEG data is shown below. It is a matrix with dimension of sensors by time. **b,** The transitions of interests are shown, with the red and blue entries indicating transitions in forward and backward direction respectively. **c,** The first step of TDLM is to construct decoding models of states from the task data, and (**d**) then transform the data (e.g., resting-state) from sensor space to the state space. TDLM works on the decoded state space throughout. **e,** The second step of TDLM is to quantify the temporal structure of the decoded states using multiple linear regressions. The first level GLM results in a state*state regression coefficient matrix at each time lag, i.e., β. **f,** The second-level GLM, this coefficient matrix is projected onto the hypothesized state transition matrix (black entries), to give a single measure of sequenceness. Repeating this process for the number of time lags of intertest, generate Z, for example, Z_F, the estimated sequenceness of forward transitions of interest over all computed time lags (solid black line). g, The third step of TDLM is statistical inference. Statistical significance is tested using a nonparametric state permutation test by randomly shuffling the transition matrix of interest (in grey). To control for multiple comparisons, the permutation threshold is defined as the 95th percentile of all shuffles on the maximum value over all tested time lags.

2.2.2 TDLM steps in detail

2.2.2.1 Getting the states

As described above, the input to TDLM is a set of time series of decoded neural representations, or states. Here we give three examples of specific state spaces (X, with dimension of time by states) that we have worked with using TDLM.

2.2.2.2 States as sensory stimuli

The simplest case, perhaps, is to define a state in terms of a neural representation of sensory stimuli, e.g., face, house. To obtain their neural representation, we present stimuli in a randomized order at the start of a task, while whole-brain neural activity is recorded by a non-invasive neuroimaging method, e.g., MEG or EEG. We then train a supervised decoding model to map the pattern of recorded neural activity to the presented image (Figure 2.2). This could be any of the multitude of available decoding models. For simplicity we have used a logistic regression model throughout.

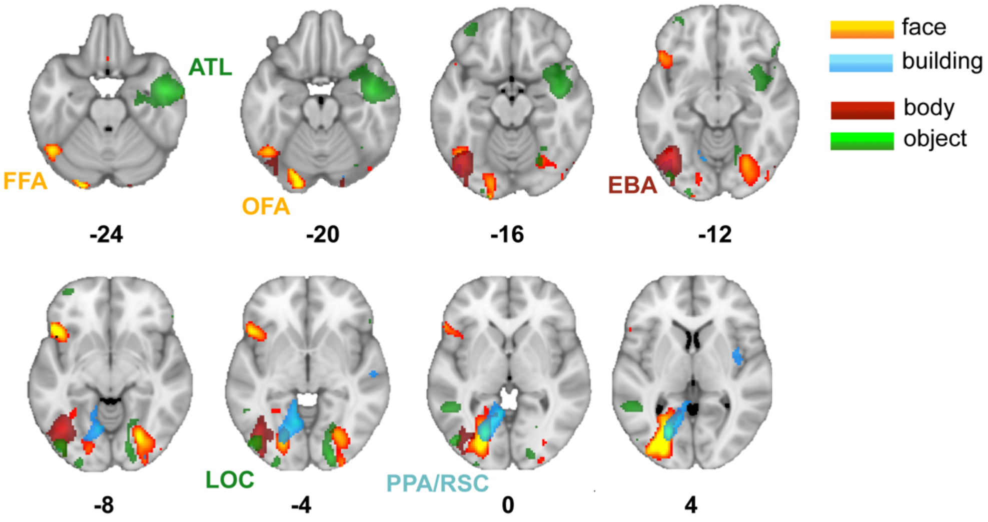


Figure 2.2 Source localization of stimuli evoked neural activity

States are defined as stimuli evoked neural activity. The classifiers are trained at 200 ms post-stimulus onset. For example, take the stimuli as consisting faces, buildings, body parts, and objects. Source localizing the evoked neural activity, we find expected activation patterns for the 4 stimuli based on literature. For faces, activation peaked in a region roughly consistent with the fusiform face area (FFA) as well as the occipital face area (OFA). Activation for building stimuli was located between the well-known parahippocampal place area (PPA) and the retrosplenial cortex (RSC), a region also known to respond to scene and building stimuli. Activation for body part stimuli was in a region consistent with the extrastriate body area (EBA). Activation for objects was in a region consistent with the object-associated lateral occipital cortex (LOC) as well as an anterior temporal lobe (ATL) cluster that may relate to conceptual processing of objects. Individual category maps thresholded to display localized peaks for illustration. This is adapted from Wimmer, et al. [58]. Full unthresholded maps can be found at https://neurovault.org/collections/6088/.

In MEG/EEG, neural activity is recorded by multiple sensor or channel arrays on the scalp. The sensor arrays record whole-brain neural activity at millisecond temporal resolution. To avoid a potential selection bias (given the sequence is expressed in time), we choose to use the whole brain sensor activity at a single time point (i.e., spatial feature) as the training data fed into classifier training.

Ideally, we would like to select a time point where the neural activity provides the most truthfully read-out. This can be indexed as the time point that gives the peak decoding accuracy. If the state is defined by the sensory feature of stimuli, we can use a classical leave-one-out cross-validation scheme to determine the ability of classifiers to generalize to unseen data of the same stimulus type (decoding accuracy) at each time point (see Appendix 2 for its algorithm box). This cross-validation scheme is asking whether the classifier trained on the sensory feature can be used to classify the unseen data of same stimuli (Figure 2.3a, b). After we have identified the peak time point based on the cross validation, we can train the decoding models based on the sensor data at that given time.

Specifically, let's denote the training data, M, with dimension of number of observations, b, by number of sensors, s. The label, Y, with dimension of b by 1. The job here is to obtain the classifier weights, W, so that $\mathrm{Y} \approx \sigma(\mathrm{MW})$. σ is the logistic sigmoid function.

Normally we apply L1 regularization on the inference of weights (we will detail the reasons in later section "Regularization"):

$$\mathrm{W} = \underset{W}{\operatorname{argmax}}[\log(\mathrm{P}(\mathrm{Y}|\mathrm{M},\mathrm{W})) + \mathrm{b}\,\lambda_{L1}\,||\,\mathrm{W}\,||_1] \tag{7}$$

After that, we can translate the data at testing time (e.g., during rest), R, from sensor space to the decoded state space:

$$\mathrm{X} = \sigma(\mathrm{RW}) \tag{8}$$

R is the testing data, with dimension of time by sensors. X is the decoded state space, with dimension of time by states.

2.2.2.3 States as abstractions

As well as sequences of sensory representations, it is possible to search for replay of more abstract neural representations, within the constraint that we can build a decoder for them. Such abstractions might be associated with the presented image (e.g., mammal vs fish), in which case the analysis can proceed as above simply by swapping categories for images.

A more subtle example, however, is where the abstraction has to do with the sequence or graph itself. For example, one representation of interest might be whatever is common at a particular location in space but invariant to what sensory stimuli are present at that location [59]. A related type of abstraction corresponds to the position of an item in a sequence, invariant to which actual item is in that very position [54,60].

We need to exercise care when setting up cross-validation schemes for training "abstract" classifiers, because we don't want the "abstract" classifier to capitalize on common sensory features. Otherwise, we might report false positive sequences of abstract codes when in fact there is only sequence for sensory information (Figure 2.4). This can happen if we train and test on the same sensory (as well as abstract) object. In other words, we need to ensure that there is no one-to one mapping between sensory and abstract code. To do so, we need more than one sensory exemplar for each abstract state.

If we have exemplars of N $(N > 1)$ different sensory images for each abstract state, then training can proceed in the following way. For example, the training set for the "2" decoder comprises $N - 1$ types of sensory images at position 2, leaving out all instances of one single type sensory example for cross-validation (see Appendix 2 for its algorithm box). Therefore, an above chance classification must rely on features that are shared between the N-1 sensory images and the one left-out sensory image, which is the abstract code. If there are just 2 stimuli per abstraction, we can train on one stimulus, and test on the other (and vice versa), selecting the time point that does best in this "cross-validation". This scheme therefore searches for representations that generalize over at least two stimuli that embody the same abstract meaning (Figure 2.3c).

Note, after we have identified the peak time point of the abstract code, we should still regress out the sensory information at given time before training classifiers for the abstract code. This is important because the above cross-validation scheme only tries to find the time when we have the strongest abstract code, it does *not* exclude sensory information at that time. We want to obtain a decoding model of *pure* abstract code; therefore, we have to make sure it does not contain sensory information. Fail to do so will lead to false positive (Figure 2.4).

Specifically, let's assume we have two training data, one for sensory code, D_s, one for abstract code, D_p. Critically, D_s does not contain abstract information and will be used to obtain the representation of pure sensory code, $W_{sensory}$. This can be achieved through either decoding (as described above) or simply averaging trials together to get the mean multivariate sensor pattern of given state (or de-mixed PCA) [61]. We can then regress out the contribution of sensory information from the training data of abstract code:

$$E_P = D_p - pinv(W_{sensory}) \times D_p \times W_{sensory} \tag{9}$$

After regressing out the sensory information, we can obtain the classifier weight of the abstract code using the residuals, E_P, as the training data. The analysis pipeline will be the same as in "States as sensory stimuli" to obtain the state space, X.

2.2.2.4 States as sequence events

TDLM can also be used iteratively to ask questions about the ordering of different types of replay events (Figure 2.3d). This can lead to powerful inferences about the temporal organization of replay, such as the temporal structure between sequences, or the repeating pattern of the same sequence. This more sophisticated use of TDLM merits its own consideration and is discussed below under "Sequences of sequences". The state space here, are meta-state, obtained based on the space of original states, X_{orig}. First, based on the transition of interest, T, we can obtain the projection matrix, X_{proj}:

$$X_{proj} = X_{orig} \times T \tag{10}$$

If we know the state lag within sequence, Δt (e.g., the time lag give rise to the strongest sequenceness) or have it a prior. We can obtain the time lagged matrix, X_{lag}:

$$X_{lag} = X_{orig}(t - \Delta \mathrm{t}) \tag{11}$$

Then, we can obtain state space with sequence event as states by elementwise multiply X_{proj} and X_{lag} (Figure 2.3d):

$$X = X_{lag} \;.\!* \; X_{proj} \tag{12}$$

Each element in X indicates the strength of a (pairwise) sequence at a given moment in time.

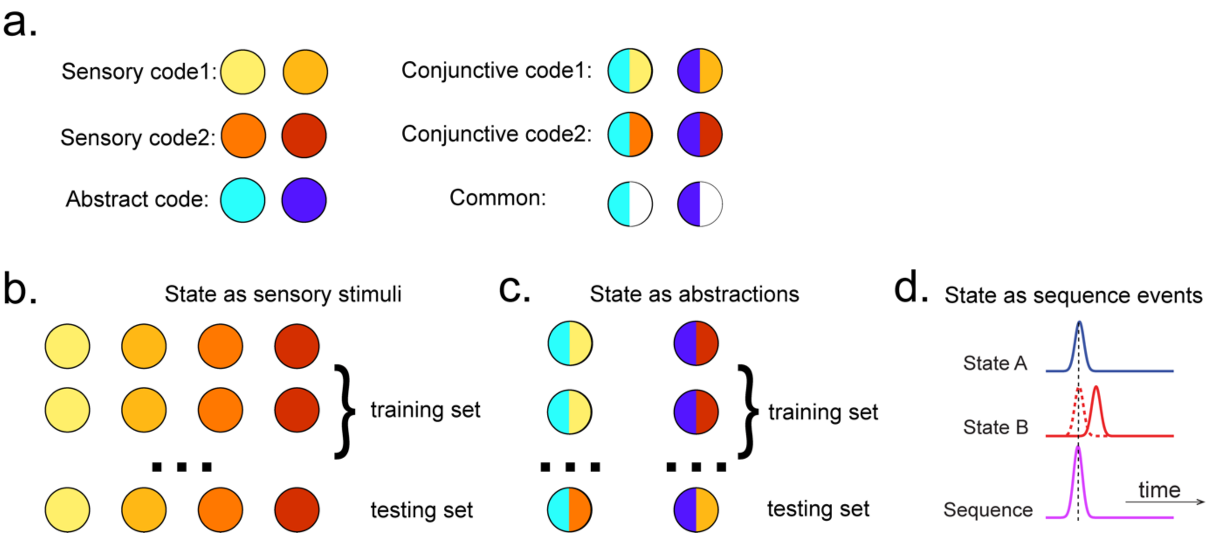


Figure 2.3 Obtaining different state spaces

a, Assuming we have two abstract codes, each abstract code has two different sensory codes (left panel). The M/EEG data corresponding to each stimulus is a conjunctive representation of sensory and abstract codes (right panel). The abstract code can be operationalized as the common information in the conjunctive codes of two stimuli that share the same abstract representation. **b,** Training decoding models for stimulus information. The simplest state is defined by sensory stimuli. To determine the best time point for classifier training, we can use a classical leave-one-out cross validation scheme on the stimuli-evoked neural activity. **c,** Training decoding models for abstract information. The state can also be defined as an abstraction. To extract this information alone, we need to avoid any 'contamination' by sensory information. We train the classifier on neural activity evoked by one stimulus and tested on another that shares the same abstract representation. If neural activity contains both a sensory and abstract code, then the only information that can generalize is the common abstract code. **d,** A state can also be defined as the sequence event itself.

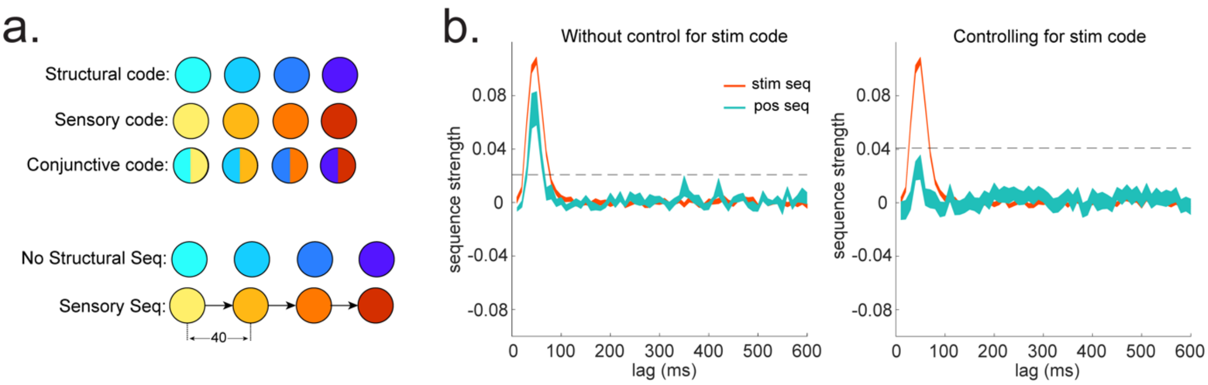


Figure 2.4 Sequences of abstract code

a, Illustration of the relationship between sensory code and (abstract) structural code. The problem is we cannot directly access structural code. We can only indirectly obtain structural code from the conjunctive code which have both sensory and structural information. As ground truth, there is a sequence of sensory code but not of structural code. b, We show in simulation the importance of controlling for sensory (stim) information, when looking for an abstract code.

2.2.3 Controlling confounds and maximizing sensitivity in sequence detection

Here, we motivate the key features of TDLM.

2.2.3.1 Temporal correlations

In standard linear methods, unmodelled temporal autocorrelations can inflate statistical scores. Techniques such as auto-regressive noise modelling are commonplace to mitigate these effects [62,63]. However, autocorrelation is a particular burden for analysis of sequences, where it interacts with correlations between the decoded neural variables.

To unpack this, consider a situation where we are testing for the sequence $X_i \rightarrow X_j$. TDLM is interested in the correlation between X_i and lagged X_j (see Equation 1). But if the X_i and X_j time series contain an autocorrelation and are also correlated with one another, then $X_i(t)$ will necessarily be correlated with $X_j(t + \Delta t)$. Hence, the analysis will report spurious sequences.

Correlations between states are commonplace. Consider representations of visual stimuli decoded from neuroimaging data. If these states are decoded using an n-way classifier (forcing exactly one state to be decoded at each moment), then the n states will be anti-correlated by construction. On the other hand, if the states are each classified against a null state corresponding to the absence of stimuli, then the n states will typically be positively correlated with one another.

Notably, in our case, because these autocorrelations are identical between forward and backward sequences, one approach for removing them is to compute a difference measure described above ($D = Z_F - Z_B$). This approach that works well and was suggested first in Kurth-Nelson, et al. [11]. However, a downside is that it prevents us from measuring forward and backward sequences independently. The remainder of this section considers alternative approaches that can allow an independent measurement of forward and backward sequences.

Moving to multiple linear regression: The spurious correlations above are induced because $X_j(t)$ mediates a linear relationship between $X_i(t)$ and $X_j(t + \Delta t)$. Hence, if we knew $X_j(t)$, we could solve the problem by simply controlling for it in the linear regression, as in Granger Causality [64]:

$$X_j(t + \Delta t) = \beta_0 + X_i(t)\beta_{ij} + X_j(t)\beta_{jj} \tag{13}$$

Unfortunately, however, we do not have access to the ground truth of X – since these variables have been decoded noisily from brain activity. Any error in $X_j(t)$ but not $X_i(t)$ will mean that the control for autocorrelation will be imperfect, leading to spurious weight on β_{ij}, and therefore spurious inference of sequences.

This problem cannot be solved without a perfect estimate of X, but it can be systematically reduced until negligible. It turns out the necessary strategy is simple. We do not know ground truth $X_j(t)$, but what if we knew a subspace that included estimated $X_j(t)$? If we controlled for that whole subspace, we would again be protected. We can get closer and closer to this by including further co-regressors that are themselves correlated with estimated $X_j(t)$ with different errors from ground truth $X_j(t)$. The most straightforward approach is to include the other states of $X(t)$, each of which has different errors, leading to the multiple linear regression of Equation 2.

Figure 2.5a shows this method applied to the same simulated data whose correlation structure induces false positives in the simple linear regression of Equation 1, and by the same logic, so also in cross correlation. This is why studies based solely on cross-correlation[53,65] cannot look for sequenceness in forward and backward direction separately, but have to rely on their asymmetry. The multiple regression accounts for the correlation structure of the data and allows for correct inferences to be made. Unlike the simple subtraction method proposed above (Figure 2.5a, left panel), the multiple regression permits a separate inference on forward and backward sequences.

Oscillations and long timescale autocorrelations are another potential confound. Equation 2 performs multiple regression, regressing each $X_j(t + \Delta t)$ onto each $X_i(t)$ whilst controlling for all other state estimates at time *t*. This method works well when spurious relationships between $X_i(t)$ and $X_j(t + \Delta t)$ are mediated by the subspace spanned by the other estimated states at time t (in particular $X_j(t)$). One situation in which this assumption might be challenged is when replay is superimposed on large scale neural oscillations. For example, during rest with eyes closed (which is often of interest in replay analysis), MEG and EEG data often express a large alpha rhythm, at around 10Hz.

If all states experience the same oscillation at the same phase, the approach correctly controls false positives. The oscillation induces a spurious correlation between $X_i(t)$ and $X_j(t + \Delta t)$ but, as before, this spurious correlation is mediated by $X_j(t)$.

However, this logic fails when states experience the oscillation at different phases. This scenario may occur, for example, if there are travelling waves in cortex [66,67], because different sensors will experience the wave at different times, and different states have different contributions from each sensor. This is different from detecting replay in rodents. The state space here is the decoded time by state matrix, like the posterior estimated posterior locations in rodents. But unlike the rodent data, we cannot obtain this matrix from spikes which do not have background oscillation, instead we decode this matrix from the MEG sensor data recorded during rest. Those MEG sensors can be seen as measures of local field potential, which do contain background oscillations. It is dominantly alpha during rest in humans.

In this case, $X_i(t)$ predicts $X_j(t + \Delta t)$ over and above $X_j(t)$. To see this, consider the situation where Δt is $\frac{1}{4}$ τ (where τ is the oscillatory period) and the phase shift between $X_i(t)$ and $X_j(t)$ is pi/2. Now every peak in $X_j(t + \Delta t)$ corresponds to a peak in $X_i(t)$ but a zero of $X_j(t)$.

To combat this problem, we can include phase shifted versions/more timepoints of $X(t)$. If a dominant background oscillation is at alpha frequency (e.g., 10Hz), neural activity at time T would be correlated with activity at time T + τ. We can control for that, by including $X(t + \tau)$, as well as $X(t)$ in the GLM (Fig. 3b). Here τ = 100 ms, if assuming the frequency is 10Hz. Applying this method to the real MEG data during rest, we see much diminished 10Hz oscillation in sequence detection during rest [54].

2.2.3.2 Spatial correlations

As mentioned above, correlations between decoded variables occur commonly. The simplest type of decoding model is a binary classifier that maps brain activity to one of two states. These states will, by definition, be perfectly anti-correlated. Conversely, if separate classifiers are trained to distinguish each state's representation from baseline ("null") brain data, then the states will often be positively correlated with each other.

Unfortunately, positive or negative correlation between states reduces the sensitivity of sequence detection, because it is difficult to distinguish between states within the sequence: collinearity impairs estimation of β in Equation 2. In Figure 2.5c, we show in simulation that the ability to detect real sequences goes down as the absolute value of spatial correlation goes up. We took the absolute value here because the direction of correlation is not important, only the magnitude of the correlation matters.

Ideally, the state decoding models should be as independent as possible. We have suggested the approach of training models to discriminate one state against a mixture of other states and against null data [53,54]. The mixture ratio can be adjusted. Adding more null data causes the states to be positively correlated with each other, while less null data leads to negative correlation. We adjust the ratio to bring the correlation between states as close to zero as possible. In Figure 2.5d, we show in simulation the benefit this has for sequence detection. An alternative method is penalizing covariance between states in the classifier's cost function [68].

2.2.3.3 Regularization

A key parameter in training high dimensional decoding models is the degree of regularization. In sequence analysis, we are often interested in spontaneous reactivations of state representations – as in replay. However, our decoding models are typically trained on stimulus-evoked data, because this is the only time at which we know the ground truth of what is being represented. This poses a challenge in so far as the models best suited for decoding evoked activity at training may not be best suited for decoding spontaneous activity at subsequent test.

During classifier training, we can impose L1, L2 constraints over the inference of classifier coefficients, W. This amount to finding the coefficients, W, that maximise the likelihood of the data observations, under the constraint imposed by the prior term. L1 regularization can be phrased as maximising the likelihood subject to a regularisation penalty on the L1 norm of the coefficient vector:

$$\mathrm{W} = \underset{W}{\operatorname{argmax}}[\log(\mathrm{P}(\mathrm{Y}|\mathrm{M},\mathrm{W})) + \mathrm{b}\,\lambda_{L1}\,||\,\mathrm{W}\,||_1] \tag{14}$$

While L2 regression can be viewed as a problem of maximizing the likelihood subject to a regularization penalty on the L2 norm of the coefficient vector:

$$\mathrm{W} = \underset{W}{\operatorname{argmax}}\left[\log\left(\mathrm{P}(\mathrm{Y}|\mathrm{M},\mathrm{W})\right) + \mathrm{b}\,\lambda_{L2}\,||\,\mathrm{W}\,||_2\right] \tag{15}$$

Where M is the task data, with dimension of number of observations, b, by number of sensors, s. Y is the label of observations, a vector with dimension of b by 1. $\mathrm{P}(\mathrm{Y}|\mathrm{M},\mathrm{W}) = \sigma(\mathrm{MW})$, and σ is the logistic sigmoid function.

We find that L1 weight regularization outperforms L2 regularization in detecting sequences (Figure 2.5e). In this simulation, we specify the number of sequences to be inserted in the resting state data. The ground truth sequence is with 40 ms time lag. The beta estimate of sequence strength at 40 ms is positively related to the number of sequences. A higher sequenceness value indicates higher sensitivity in detecting the ground truth sequence. In Figure 2.5e, the green dot there is the sequence detection ability without any regularization but with everything else being equal. This can be seen a baseline for measuring the performance of L1 and L2 regularization. We can see L1 regularization (red dots) increases sequence detection ability compared to baseline (green dot), while the L2 regularization (blue dots) does not. The L1, and L2 regularization-based sequence results are plotted also against the absolute spatial correlation. We can see the reason L1 achieve the higher sequence detection ability is because it reduces the covariances between the classifiers.

In addition to minimizing spatial correlation as discussed above. It is also shown that the L1-induced sparsity encodes weaker assumptions about background noise distributions into the classifiers as compared to L2 regularization [69]. This may be of special interest to researchers who want to detect replay during sleep – where the use of sparse classifiers would be helpful when applied to sleep data, as background noise distributions are likely to differ substantially from the (awake state) training data.

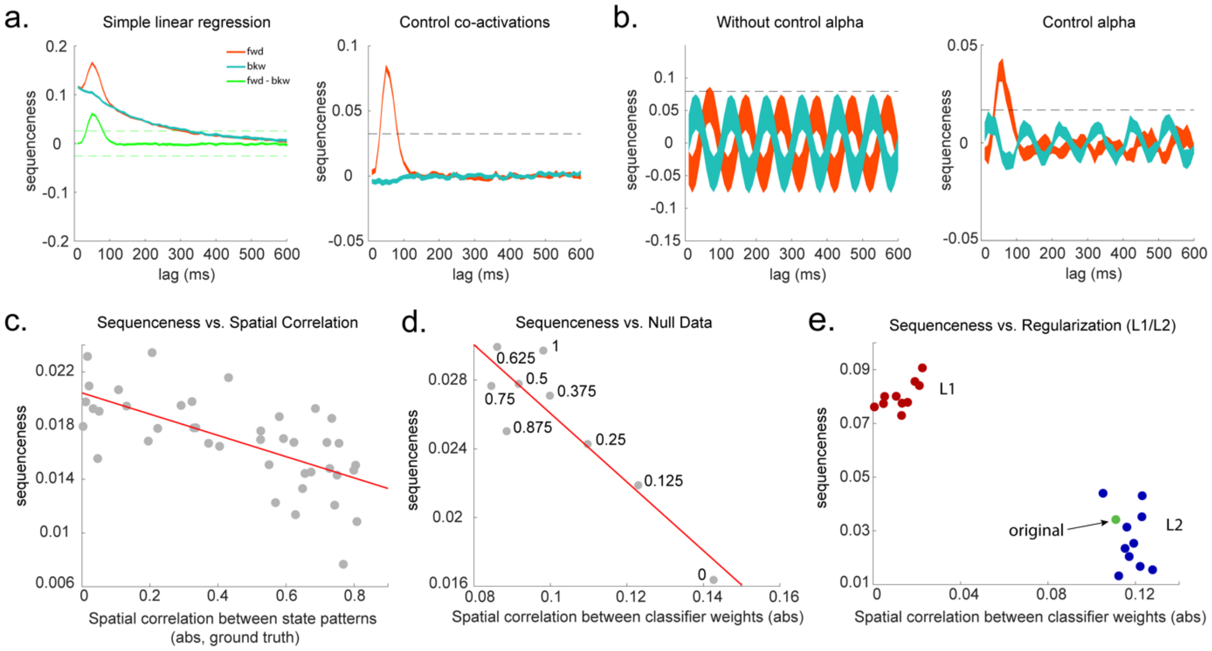


Figure 2.5 Effects of temporal, spatial correlations, and classifier regularization on TDLM

a, Simple linear regression or cross-correlation approach relies on the asymmetry of forward and backward transitions; therefore, subtraction is necessary (left panel). TDLM instead relies on multiple linear regression. TDLM can assess forward and backward transitions separately (right panel). **b,** Background alpha oscillations, as seen during rest periods, can reduce sensitivity of sequence detection (left panel), controlling alpha in TDLM helps recover the true signal (right panel). **c,** The spatial correlation between the sensor weights of decoders for each state reduces sensitivity in sequence detection. This suggests reducing overlapping patterns between states is important for sequence detection. **d,** Adding null data to a training set can increase the sensitivity of sequence detection by reducing the spatial correlations of the trained classifier weights. Here the number indicates the ratio between null data and task data. "1" means the same amount of null data and the task data. "0" means no null data is added for training. **e,** L1 regularization helps sequence detection by reducing spatial correlations (all red dots are L1 regularization with a varying parameter value), while L2 regularization does not help sequenceness (all blue dots are L2 regularization with a varying parameter value) as it does not reduce spatial correlations of the trained classifiers compared to the classifier trained without any regularization (green point).

2.2.4 Statistical inference

So far, we have shown how to quantify sequences in representational dynamics. An essential final step involves assessing the statistical reliability of these quantities.

All the tests described in this section evaluate the consistency of sequences *across subjects*. This is very important, because even in the absence of any real sequences of task-related representations, spontaneous neural activity is not random but often follows repeating dynamical motifs [70]. Solving this problem requires a randomized mapping between the assignment of physical stimuli to task states. This can be done across subjects, permitting valid inference at the group level.

At the group level, the statistical testing problem can be complicated by the fact that sequence measures do not, in general, follow a known distribution. Additionally, if the state-to-state lag of interest (Δt) is not known a priori, it will be necessary to perform tests at multiple lags, creating a multiple comparisons problem over a set of tests with complex interdependencies. In this section we discuss inference with these issues in mind.

2.2.4.1 Distribution of sequenceness at a single lag

If the state-to-state lag of interest (Δt) is known a priori then the simplest approach is to compare the sequenceness against zero, for example using either a signed-rank test, or one-sample *t* test (assuming Gaussian distribution). Such testing assumes that the data is centred on zero, if there were no real sequences. We show this approach is safe in both simulation (assuming no real sequences) and when using real MEG data where we know there are no sequences.

In simulation, we assume no real sequences, but state time courses are autocorrelated. At this point, there is no systematic structure in the correlation between the neuronal representations of different states (see later for this consideration). We then simply select the 40 ms time lag and compare its sequenceness to zero, using either a signed-rank test or one-sample *t* test. We can compare false positive rates predicted by the statistical tests with false positive rates measured in simulation (Figure 2.6a). Here we see the empirical false positives are well predicted by theory.

We can also test this on real MEG data. In Liu, et al. [54] we had one condition where we measured resting activity before subjects saw any stimuli. Therefore, by definition these stimuli could not embody replay of states of interest, but we can use the classifiers from these state/stimuli (measured later) to test the false positive performance of statistical tests of replay. We note this is different to the preplay phenomena observed in rodent literature. In rodent, the preplay happens before the rodent enters a novel maze, where the suggestion is that this is due to an anatomical defined canonical dynamic in the hippocampus [18,71]. Crucially, the transitions in the spatial space are fixed, and there are (nearly) one-to-one mapping between place cells and locations in the space. Both of these are not true in the target MEG data. In this MEG experiment, we analyse resting state data before stimuli presentation. In this case, it is like the preplay experiment, but unlike the rodent experiment, the mapping between stimuli and state are randomized across subjects, and on average we are looking for consistent state-to-state transitions. Even there is a consistent dynamic between stimuli-related processing, e.g., face -> house, they will indicate different states in different subjects, so that one would expect this *stimuli* preplay will not contribute to the *stimuli-defined state* preplay.

To obtain many examples, we randomly permute the 8 different stimuli 10,000 times and then compare sequenceness (at 40 ms time lag) to zero using either signed rank test or one-sample *t* test across subjects. Again, predicted and measured false positive rates match well (Figure 2.6b, left panel). This holds true across all computed time lags (Figure 2.6b, right panel).

An alternative to making assumptions about the form of the null distribution is to compute an empirical null distribution by permutation. Given we are interested in the sequence of states over time, one could imagine permuting either state identity or time. However, permuting time uniformly typically leads to a very high incidence of false positives, as time is not exchangeable under the null hypothesis (Figure 2.6c, blue colour). This false positive also exists if we circular shift the time dimension of each state rather than randomly permutating it. Permuting time destroys the temporal smoothness of neural data, creating an artificially narrow null distribution [53,54]. State permutation, on the other hand, only assumes state identities are exchangeable under the null hypothesis, while preserving the temporal dynamics of the neural data, represents a safer statistical test that is well within 5% false positive rate (Figure 2.6c, purple colour).

2.2.4.2 Correcting for multiple comparisons

If the state-to-state lag of interest is not known, then we have to search over a range of time lags. As a result, this will create a multiple comparison problem. Unfortunately, we don’t yet have a good parametric method to control for multiple testing over a distribution. It is possible to use methods that exploit the properties of Gaussian Random Fields, as is common in the analysis of fMRI [72], but we have not evaluated this approach. We could also use a Bonferroni correction, but there is a limitation due to an assumption that each computed time lag is independent is likely false and overly conservative.

As an alternative we recommend relying on state-identity based permutation. To control the family wise error rate (assuming $\alpha = 0.05$), we want to ensure that there is a 5% probability of getting the tested sequenceness strength (S_{test}) or bigger by chance in *any* of the multiple tests. We therefore need to know what fraction of the permutations give S_{test} or bigger in any of *their* multiple tests. If any of the sequenceness scores in each permutation exceed S_{test}, then the *maximum* sequenceness score in the permutation will exceed S_{test}, so it is sufficient to test against the maximum sequenceness score in the permutation. The null distribution is therefore formed by first taking the peak of sequenceness across all computed time lags of each permutation. This is the same as approach as is used for family-wise error correction for permutations tests in fMRI data [73], and in our case it is shown to behave well statistically (Figure 2.6d).

2.2.4.3 What to permute

We can choose which permutations to include in the null distribution. For example, consider a task with two sequences, $Seq1$: $A \rightarrow B \rightarrow C \rightarrow D$, and $Seq2$: $E \rightarrow F \rightarrow G \rightarrow H$. We can form the null distribution either by permuting all states (e.g., one permutation might be: E→ $F \rightarrow A \rightarrow B$, H→ $C \rightarrow E \rightarrow D$), as was performed in Kurth-Nelson, et al. [53]. Alternatively, we can form a null distribution which only includes transitions between states in different sequences (e.g., one permutation might be: D→ $G \rightarrow A \rightarrow E$, H→ $C \rightarrow F \rightarrow B$), as was performed in Liu, et al. [54]. In each case, permutations are equivalent to the test data under the assumption that states are exchangeable between positions and sequences. The first approach has the advantage of many more possible permutations, and therefore may make more precise inferential statements in the tail. The second may be more sensitive in the presence of signal, as the null distribution is guaranteed not to include permutations which share any transitions with the test data (Figure 2.6e). For example, in the Figure 2.6e, the blue swaps are the permutations that only exchange state identity across sequences, as in Liu, et al. [54]; while the red swaps are the permutations that permit all possible state identity permutations, as in Kurth-Nelson, et al. [53]. There will be many more different state permutations in red swaps than in blue swaps. We can make different levels of inferences by controlling the range of the null distributions in the state permutation tests.

This state identity-based permutation is similar to shuffling the rate maps of each place cell in rodent ephys analysis. Effectively, they both make a null distribution where states (positions) are exchangeable at the decoded state space. We cannot do rate map like shuffling in MEG data, because the state (with analogy to position) and sensors (with analogy to place cells) mappings are not one-to-one. Instead, the state patterns are multivariate in MEG. The state-sensors shuffling in MEG analysis will not make the same null distribution (states are exchangeable), it instead will make the state decoding nosier.

2.2.4.4 Cautionary note on exchangeability of states after training

Until now, all tests have assumed that state identity is exchangeable under the null hypothesis. Under this assumption, it is safe to perform state-identity based permutation tests on Z_F and Z_B. In this section, we consider a situation where this assumption is broken.

More specifically, we are considering a situation where the neural representation of state A and B are related in a systematic way or, in other words, the classifier on state A is confused with state B, and we are testing sequenceness of $A \rightarrow B$. Crucially, to break the exchangeability assumption, representations of A and B have to be systematically more related than other states, e.g., A and D. This cannot be caused by low level factors (e.g., visual similarity) because states are counterbalanced across subjects, so any such bias would cancel out at the population level. However, such a bias might be *induced* by task training.

In this situation, it is, in principle, possible to detect sequenceness of $A \rightarrow B$, even in the absence of real sequences. In the autocorrelation section above, we introduced protections against the interaction of state correlation with autocorrelation. These protections can fail in the current case as we cannot use other states as controls (as we do in the multiple linear regression), because A has systematic relationship with B, but not with other states. State permutation will not protect us from this problem because state identity is no longer exchangeable.

Is this a substantive problem? After extensive training, behavioral pairing of stimuli can indeed result in increased neuronal similarity [74,75]. These early reports relate to lengthy training in monkeys. More recent studies have shown that an induced representational overlap can be seen in human imaging within a single day [57,76,77]. However, when analyzed across the whole brain, such representational changes tend to be localized to discrete brain regions [78,79], and as a consequence are likely to have limited impact on whole brain decodeability.

Whilst we have not yet found a simulation regime in which false positives are found (as opposed to false negatives), there exists a danger in cases where, by experimental design, the states are not exchangeable.

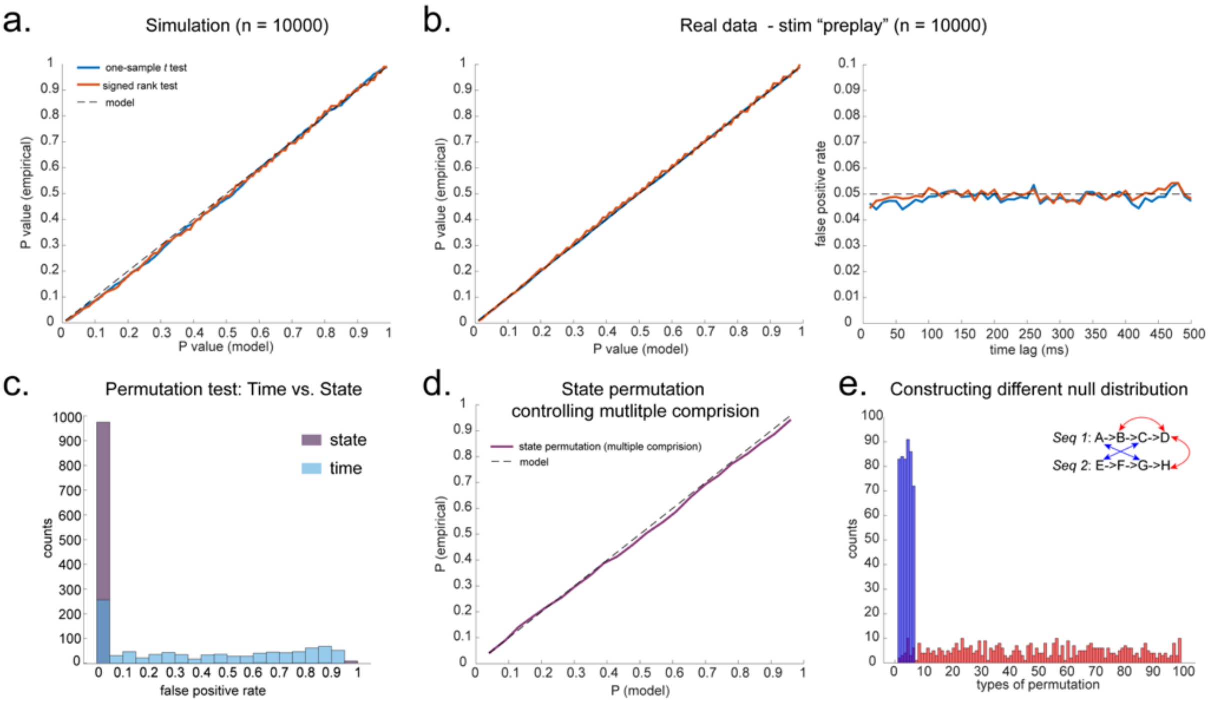


Figure 2.6 Statistical inference

a, P-P plot of one-sample *t* test (blue) and Wilcoxon signed rank test (red) against zero. This is done in simulated MEG data assuming auto-correlated state time courses but no real sequences. In each simulation, the statistics are done only on sequenceness at 40 ms time lag, across 24 simulated subjects. There are 10,000 simulations. **b,** We have also tested the sequenceness distribution on the real MEG data. The state identity is randomly shuffled 10,000 times to construct the null distribution. **c,** Time-based permutation test tends to give high false positive, while state identity-based permutation does not. This is done in simulation assuming no real sequences (n=1000). **d,** P-P plot of state identity-based permutation test over peak sequenceness is shown. To control for multiple comparisons, the null distribution is formed taking the maximal absolute value over all computed time lags within a permutation, and the permutation threshold is defined as the 95% percentile over permutations. In simulation, we only compared the max sequence strength in the data to this permutation threshold. **e,** Blue are the permutations that only exchange state identity across sequences. Red are the permutations that permit all possible state identity permutations. The X axis is the different combinations of the state permutation. It is sorted so that the cross-sequence permutations are in the beginning.

2.2.5 Extensions to TDLM

TDLM can be used iteratively. Two extensions of TDLM of particular interest are: Multi-step sequences and Sequence of sequences. The former asks about consistent regularity among multiple states, the latter ask about the hierarchical structure of state reactivation, not only within but between sequences.

2.2.5.1 Multi-step sequences

So far, we have introduced methods for quantifying the extent to which the state-to-state transition structure in neural data matches a hypothesized task-related transition matrix. An important limitation of these methods is that they are blind to hysteresis in transitions. In other words, they cannot tell us about multi-step sequences. In this section, we describe a methodological extension to measure evidence for sequences comprising more than one transition: for example, $A \rightarrow B \rightarrow C$.

The key ingredient is controlling for shorter sub-sequences (e.g., $A \rightarrow B$ and $B \rightarrow C$), in order to find evidence unique to the multi-step sequence of interest.

Assuming constant state-to-state time lag, Δt, between A and B, and between B and C. We can create new state space AB, by shifting B up Δt, and elementwise multiply it with state A. This new state AB measure the reactivation strength of $A \rightarrow B$, with time lag Δt. In the same way, we can create new state space, BC, AC, etc. Then we can construct the same first level GLM on the new state space. For example, if we want to know the evidence of $A \rightarrow B \rightarrow C$ at time lag Δt. We can regress AB onto state time course C, at each Δt (cf. Equation 1). But we want to know the unique contribution of AB to C. More specifically, we want to test if the evidence of $A \rightarrow B \rightarrow C$ is stronger than $X \rightarrow B \rightarrow C$, where X is any state but not A. Therefore, similar as Equation 2, we want to control CB, DB, when looking for evidence of AB of C. Applying this method, we show TDLM successfully avoids false positives arising out of strong evidence for shorter length (see simulation results in Figure 2.7a, see results obtained on human neuroimaging data in Figure 2.7b). This process can be generalized to any number of steps.

TDLM in current form assuming a constant state-to-state time lags intra-sequence. If one wants to have variability between state transitions, TDLM can still cope, but not very elegantly. Assuming there is a three states sequence, $A \rightarrow B \rightarrow C$, with intra-sequence variance. TDLM will need to test all possible combinations of state-to-state time lags in $A \rightarrow B$ and $B \rightarrow C$. If there are n number of time lag of interest in either of the two transitions, TDLM will have to test n^2 possible time lag combinations. This is a big search space and will increase exponentially as a function of the length of sequence.

We also note this analysis is different from a typical rodent replay analysis which assesses the overall evidence of the sequence length [19,56]. TDLM is asking do we see more evidence of A->B->C, above and beyond B->C, for example. However, if the main question of interest is "do we have evidence of A->B->C in general", as is normally the case in the rodent replay analysis [19,56], then we should not control for shorter lengths, we can instead average the evidence together, as have done in Kurth-Nelson, et al. [53].

2.2.5.2 Sequence of sequences

We have so far detailed the use of either sensory or abstract representations as states in TDLM. We now take a step further and use sequences themselves as states. With this kind of hierarchical analysis, we can search for sequences of sequences. This is useful because it can reveal the temporal structure not only within sequence, but also between sequences. The organization between sequences is of particular interest for revealing neural computations. For example, the forward and backward search algorithms hypothesized in planning and inference [80] can be cast as sequences of sequences problem: the temporal structure of forward and backward sequence. This can be tested by using TDLM iteratively.

As yet little human neural data is available on the organization of sequences. Interestingly, one can think of theta sequence, a well-documented phenomenon during rodent spatial navigation [39,42,55], as a neural sequence repeating itself in theta frequency (6 - 12 Hz). We will show TDLM is able to detect this well-known phenomenon.

To look for sequences between sequences we need first to define sequences as new states. To do so, the raw state course, for example, state B needs to be shifted up by the empirical within-sequence time lag Δt (determined by the two-level GLM described above), to align with the onset of state A, if assuming sequence $A \rightarrow B$ exist (at time lag Δt). Then, we can elementwise multiply the raw state time course A with the shifted time course B, resulting in a new state AB (Figure 2.3d). Each entry in this new state time course indicates the reactivation strength of sequence AB at a given time.

After that, the general two-level GLMs framework still applies, but with one important caveat. The new sequence state (e.g., AB) is defined based on the original states (A and B), and we are now interested in the reactivation regularity, i.e., sequence, between sequences, rather than the original states. We should therefore control for the effects of the original states. Effectively, this is like controlling for main effects (e.g., state A and shifted state B) when looking for their interaction (sequence AB). TDLM achieves this by putting time lagged original state regressors A, B, in addition to AB, in the first level GLM sequence analysis.

Specifically, let's assume the sequence state matrix is X_{seq}, after transforming the original state space to sequence space based on the empirical within-sequence time lag Δt_w. Each column at X_{seq} is sequence state, denoted by S_{ij}, which indicates the strength of sequence i -> j reactivation. The raw state i is X_i, and the shifted raw state j is X_{jw} (by time lag Δt_w).

In the first level GLM, TDLM ask for the strength of unique contribution of sequence state S_{ij} to S_{mn} while controlling for original states (X_i and X_{jw}). For each sequence state ij, at each possible time lag Δt, TDLM estimated a separate linear model:

$$S_{mn} = X_i(\Delta t)\beta_i + X_{jw}(\Delta t)\beta_j + S_{ij}(\Delta t)\beta_{ij}(\Delta t) \tag{16}$$

Repeat this process for each sequence state separately at each time lag, resulting a sequence matrix β_{seq}.

In the 2nd level GLM, TDLM asks how strong the evidence of sequence of interest is compared to sequences that have the same starting state or end state at each time lag. This 2nd level GLM will be the same as the equation 5, but with additional regressors to control for sequences that share the same start or end state.

In simulation we demonstrate, applying this method, that TDLM can uncover hierarchical temporal structure: state A is temporally leading state B with 40 ms lag, and the sequence A->B tends to repeat itself with a 140 ms gap (Figure 2.7c). On real rodent hippocampal electrophysiological recording, we replicate the well-known theta sequence - neural sequence repeating itself in theta frequency (Figure 2.7d, see detailed analysis on this rodent data in Methods).

In addition to looking for temporal structure of the same sequence, this method is equally suitable when searching for temporal relationship between difference sequences in a general form. For example, assuming two different types of sequences, one sequence type has a within-sequence time lag at 40 ms; while the other has a within-sequence time lag at 150 ms; and there is a gap of 200 ms between the two types of sequences (Figure 2.8a) (these time lags are set arbitrarily for illustration purposes. TDLM captures accurately the dynamics both within and between the sequences (Figure 2.8b, c), supporting a potential for uncovering temporal relationships between sequences in general under the same framework.

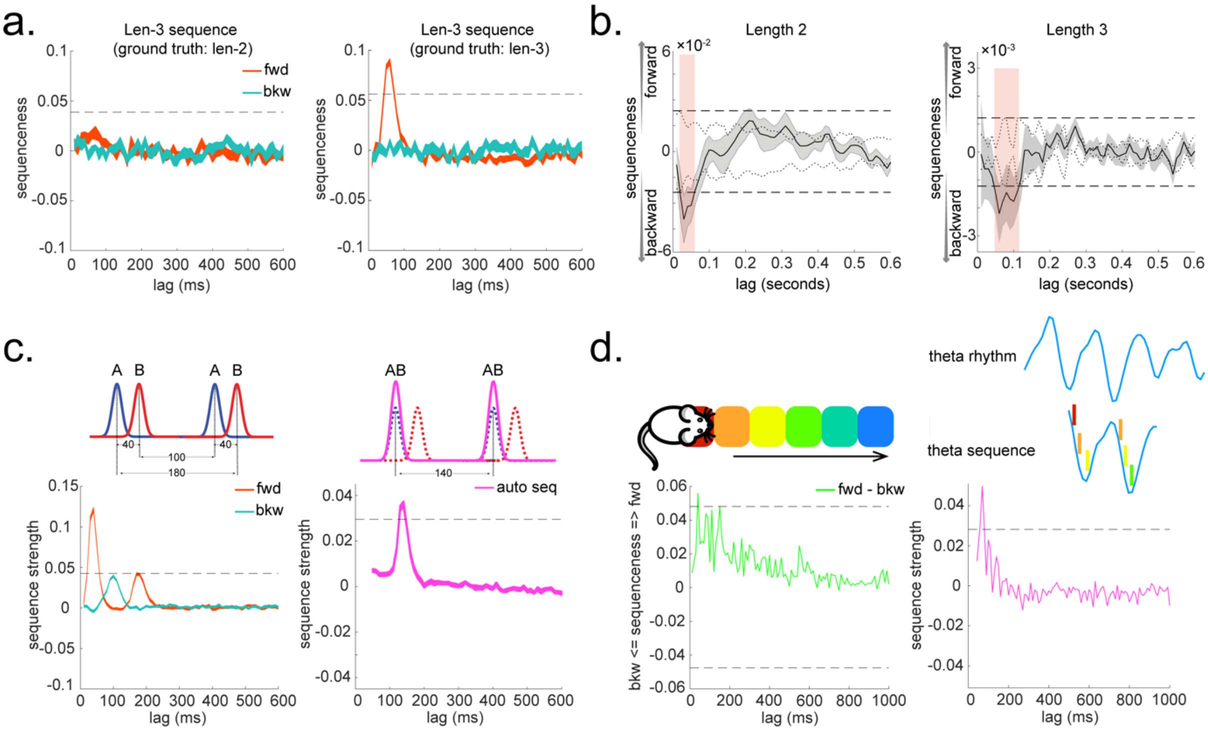


Figure 2.7 Extension to TDLM: Multi-step sequences and Sequence of sequences.

a, TDLM can quantify not only pair-wise transition, but also longer length sequences. It does so by controlling for evidence of shorter length to avoid false positive. **b,** Method applied to human MEG data, incorporating control of both alpha oscillation and co-activation for both length-2 and length-3 sequence length. Dashed line indicates the permutation threshold. This is adapted from Liu, et al. [541]. **c,** TDLM can also be used iteratively to capture the repeating pattern of sequence event itself. Illustration in the top panel describes the ground truth in the simulation. Intra-sequence temporal structure (right) and inter-sequence temporal structure (right) can be extracted simultaneously. **d,** On a real rodent hippocampal electrophysiological dataset, TDLM revealed the well-known theta sequence phenomena during active spatial navigation.

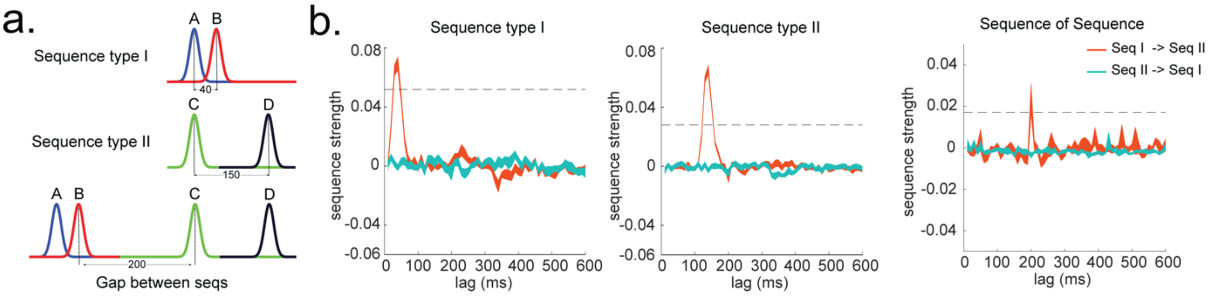


Figure 2.8 Temporal structure between and within different sequences

a, Illustration of two sequence types with different state-to-state time lag within sequence, and a systematic gap between the two types of sequences. **b,** TDLM can capture the temporal structures both within (left panel) and between (right panel) the two sequence types.

2.2.6 Source localization

Uncovering the temporal structure of neural representation is important, but one might also want to ask where in the brain the sequence is generated. Rodent electrophysiology research focuses mainly on hippocampus when searching for replay. One advantage of whole-brain non-invasive neuroimaging over electrophysiology (despite many known disadvantages, including poor anatomical precision, low signal-noise ratio) is its ability to look for neural activity in other brain regions. Ideally, we would like a method that is capable of localizing sequences of more abstract representation in brain regions beyond hippocampus [54].

We want to identity the time when a given sequence is very likely to happen. We can achieve this, by transforming from the space of states to the space of sequence event. This will be the same computation as in section "States as sequence events". The Δt is obtained by availing of the two-level GLMs in TDLM to identify the empirical time lag that gives rise to the strongest neural sequence.

After obtaining the time course of sequence events, TDLM identifies the sequence onset by thresholding the sequence state at its high (e.g., 95th) percentile with a constraint that a sequence onset has a sequence-free time window (e.g., 100 ms) preceding it. This analysis pipeline gives a temporal stamp on the testing time (Figure 2.9a). One can therefore epoch the data based on those sequence onsets and apply temporal frequency analysis and source localization, just like on the standard task data. This approach is similar to spike-triggered averaging [81,82]. Applying this to real MEG data during rest, we can detect increased hippocampal power at 120-150 Hz, during replay onset (Figure 2.9b, c).

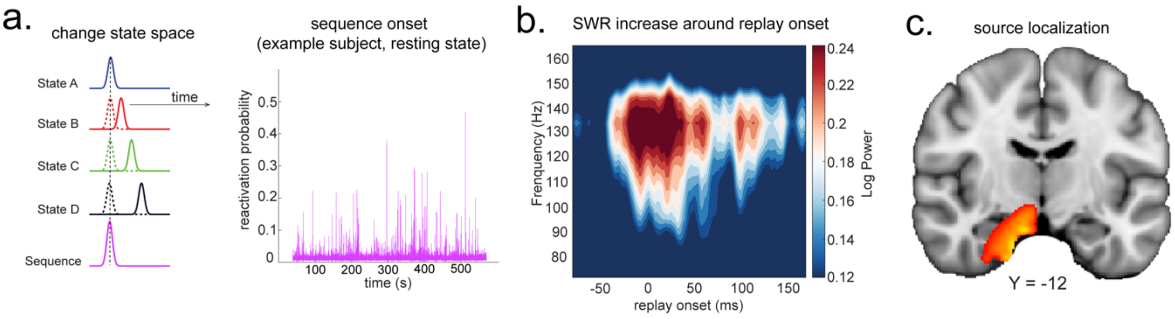


Figure 2.9 Source localization of replay onset

a, TDLM figures out the onset of sequence based on the identified optimal state-to-state time lag (left panel). Sequence onset during resting state from one example subject is shown (right panel). **b,** There was a significant power increase (averaged across all sensors), in the ripple frequency band (120-150 Hz), at the onset of replay, compared to the pre-replay baseline (100 to 50 ms before replay). c, Source localization of ripple-band power at replay onset revealed significant hippocampal activation (peak MNI coordinate: X = 18, Y = -12, Z = -27). Panel b and c are adapted from Liu, et al. [54].

2.3 Discussion

TDLM is a general analysis framework for capturing sequence regularity of neural representations. We described the application of TDLM mostly during off-task state. However, the very same analysis can be applied to on-task data, to test for cued sequential reactivation [58], or sequential decision-making [83]. It is developed on human neuroimaging data but can be applied to other data sources, including rodent electrophysiology recordings. The framework can facilitate cross-species investigations and enables investigation of phenomena that are not readily addressable in rodents [54].

The temporal dynamics of neural states have been studied previously with MEG [70,84]. Normally states are defined by common physiological features (e.g., frequency, functional connectivity) during rest, and termed resting state networks (e.g., default mode network [3]). However, these approaches remain agnostic about the *content* of neural representation. Being able to study the temporal dynamics of *representational content* permits richer investigations into cognitive processes, as neural states can be analyzed in the context of their roles with respect to cognitive tasks.

Reactivation of neural representations have also been studied previously [85] using approaches similar to the decoding step of TDLM, or multivariate pattern analysis (MVPA) [86]. This has proven fruitful in revealing mnemonic functions[77], understanding sleep[87], and decision-making[88]. However, classification alone cannot reveal the rich temporal structures of reactivation dynamics. For example, the ability to detect sequences allows us to tease apart clustered from sequential reactivation, where this may be important for dissociating decision strategies [65] and their individual differences [58,65]. Furthermore, it enables comparisons with the sequential reactivation patterns reported in rodent hippocampus [16,20], and may allow tests of neural predictions from process models such as reinforcement learning [89], which have been hard to probe previously in humans [83].

We have mainly discussed the application of TDLM on high temporal resolution neuroimaging data (e.g., MEG, see also Appendix 1 on detecting replay using EEG). Recently, sequential replay has been reported using fMRI [31]. We anticipate it will be useful to combine the high temporal resolution available in M/EEG and the spatial precision available in fMRI to probe region - specific sequential computation. Whilst related techniques are available [90], TDLM could, in principle, also be applied to fMRI data. For fMRI data, it seems it might be better to work on the certain frequency range [90], we have not explored the feature selection process so far. In future work, it will be useful to identify decoding features that are most suitable in different imaging modality.

TDLM is based on general linear models. This gives us flexibility to handle potential confounds, but only in linear fashion, it also cannot handle uncertainty in the current form. Recent success in applying latent state space model, like hidden Markov model, to detect replay in rodents [91], suggesting a Bayesian treatment of the neural dynamics may be a promising direction to explore. The ability to handle uncertainty in Bayesian treatment is also desirable. Incorporating the uncertainty in the estimate of sequenceness, could, for example, allow us to separate process noise (e.g., intrinsic variability within sequences) and measurement noise (e.g., noise in MEG recording). This could be done in building a generative model (e.g., Karman filter), a direction worth exploring in future.

TDLM enables neuroscientists to decipher rich temporal structures of neural reactivation. We believe TDLM opens doors for novel investigations of human cognition, including language, sequential planning and inference in non-spatial cognitive tasks [53,65]. It is particularly suited to test specific neural prediction from process models. Therefore, we hope TDLM can aid a synthesis between empirical and theoretical approaches in neuroscience and in so doing shed novel lights on dynamic neural computation.

2.4 Methods

2.4.1 Simulating MEG data

We simulate the data to be similar with the human MEG.

We generate ground truth multivariate patterns (over sensors) of the states. We then add random unit gaussian noise on the ground truth state patterns to form the task data. We will train a logistic regression classifier on the task data to obtain the decoding model of each of the state patterns. Later we will use this model to transform the resting-state data from sensor space (with dimension of time by sensors) to the state space (with dimension of time by states).

First, to imitate temporal autocorrelation and spatial correlation that are commonly seen in human neuroimaging data, we generate the rest data using an auto-regressive model with multivariate (over sensors) gaussian noise and adding dependence among sensors. In some simulation, we will add a rhythmic oscillation (e.g., 10Hz) on top of that.

Second, we inject sequence of the state patterns in the rest data. The sequences follow the ground truth of state transitions of interest. The state-to-state time lag is assumed to follow gamma distribution. We vary the number of sequences to be injected in the rest data to control the strength of sequences.

Lastly, we project the rest data to the decoding model of states obtained from the task data. TDLM will then work on the decoded state space.

An example of the MATLAB implementation is called "Simulate_Replay" from the GitHub link:

https://github.com/yunzheliu/TDLM

2.4.2 Human MEG dataset

Task design

Participants were required to perform a series of tasks with concurrent MEG scanning (see details in Liu, et al. [54]). The functional localizer task was performed before the main task and was used to train a sensory code for eight distinct objects. Note, the participants were provided with no structural information at the time of the localizer. These decoding models, trained on the functional localizer task, capture a sensory level neural representation of stimuli (i.e., stimulus code). Following that, participants were presented with the stimuli and were required to unscramble the "visual sequence" into a correct order, i.e., the "unscrambled sequence" based on a structural template they had learned the day before. After that, participants were given a rest for 5 mins. In the end, stimuli were presented again in random order, and participants were asked to identify the true sequence identity and structural position of the stimuli. Data in this session are used to train the structural code of the objects.

MEG data Acquisition, Pre-processing and Source Reconstruction

This is exactly the same procedure that has been reported in Liu, et al. [54]. We have copied it here for references.

“MEG was recorded continuously at 600 samples/second using a whole-head 275-channel axial gradiometer system (CTF Omega, VSM MedTech), while participants sat upright inside the scanner. Participants made responses on a button box using four fingers as they found most comfortable. The data were resampled from 600 to 100 Hz to conserve processing time and improve signal to noise ratio. All data were then high pass filtered at 0.5 Hz using a first order IIR filter to remove slow drift. After that, the raw MEG data were visually inspected, and excessively noisy segments and sensors were removed before independent component analysis (ICA). An ICA (FastICA, http://research.ics.aalto.fi/ica/fastica) was used to decompose the sensor data for each session into 150 temporally independent components and associated sensor topographies. Artefact components were classified by combined inspection of the spatial topography, time course, kurtosis of the time course and frequency spectrum for all components. Eye-blink artefacts exhibited high kurtosis (>20), a repeated pattern in the time course and consistent spatial topographies. Mains interference had extremely low kurtosis and a frequency spectrum dominated by 50 Hz line noise. Artefacts were then rejected by subtracting them out of the data. All subsequent analyses were performed directly on the filtered, cleaned MEG signal, in units of femtotesla.

All source reconstruction was performed in SPM12 and FieldTrip. Forward models were generated on the basis of a single shell using superposition of basis functions that approximately corresponded to the plane tangential to the MEG sensor array. Linearly constrained minimum variance beamforming [92], was used to reconstruct the epoched MEG data to a grid in MNI space, sampled with a grid step of 5 mm. The sensor covariance matrix for beamforming was estimated using data in either broadband power across all frequencies or restricted to ripple frequency (120-150 Hz). The baseline activity was the mean neural activity averaged over -100 ms to -50 ms relative to sequence onset. All non-artefactual trials were baseline corrected at source level. We looked at the main effect of the initialization of sequence. Non-parametric permutation tests were performed on the volume of interest to compute the multiple comparison (whole-brain corrected) P-values of clusters above 10 voxels, with the null distribution for this cluster size being computed using permutations (n = 5000 permutations).

2.4.3 Rodent hippocampal ephys for detecting theta sequence

Data description

The rodent data is collected by Héctor Penagos from Matt Wilson’s lab.

IACUC statement: Surgical procedures and behavioral testing were approved by the Committee of Animal Care at Massachusetts institute of Technology and followed US National Institute of Health guidelines (http://dspace.mit.edu/handle/1721.1/58398).

Data were collected in a spatial navigation task where the rat ran back and forth on a circular track that had a high-wall divider with reward sites on either side. The rat completed 6 rounds of run (both clockwise and counterclockwise). Fifty-three Cells were recorded in the CA1 of the hippocampus. Spiking activity was recorded at 31,250 Hz /channel. The local field potential was sampled at 2000 Hz. The position of the rat was simultaneously recorded with a sampling rate of 30 Hz. The position records were linearized for later analysis.

Preprocessing

The pre-processing steps:

1) Data are subset based on the running speed - only the time when the running speed is greater than 10 cm/s is included.
2) Putative interneurons are excluded based on their firing field – only neurons with single dominate firing field are included.
3) The running track is first linearized and then discretized into 5 cm spatial bins, for later estimation of tuning carves and Bayesian decoding analysis.
4) The time dimension is discretized into 10 ms time bins for Bayesian decoding analysis.

Theta sequences

The theta sequence analysis steps:

1) Estimation of tuning curves/rate map: The average firing rate of each place cell at each location (every 5 cm spatial bins) on the track in each running direction is estimated separately by summing the spike counts within each spatial bin and then normalized by position visited counts.
2) Bayesian decoding: We applied the standard one-step Bayesian decoding method [93]. This method uses an average rate map of each cell to estimate the probability distribution of the animal's position given a spike count vector at given time bin. We don't care about the specific running direction; we want to estimate the posterior location that does not depends on the running direction. To achieve this, we stack the two directional tuning curves into one vector for the Bayesian decoding, and later marginalized over the directions to get the posterior estimation of location probability. We choose to do so, also because it balances out the behavioral sampling experience. This will give us a posterior position probability readout at

each time bin, i.e., a decoding matrix, times*positions (states). This matrix will be used for later sequence analysis.

3) General pair-wise sequence analysis: TDLM is then run on the raw decoded probabilities (not the MAP locations). The results are shown on the left plot, Fig. 5d. This measures the average of all pairwise sequences as a function of time lag. This analysis is trying to find the time lag that give rise to the strongest sequenceness value, which will be use to change the state space from the individual position state (e.g., A, B), to position-pair states based on the lag (e.g., AB), for later analysis.
4) Repetition of sequence (theta sequence) analysis: After we have figured out the time lag give rise to the strongest pair-wise sequence, we can time shift the states in decoding matrix based on the time lag, for example, if we know position A is always activated 40 ms earlier than position B, to obtain the sequence onset probability of A->B, we can time shift the state B time course 40 ms earlier and then element-wise multiple it with the raw state A time course. We can do that for each successive position pair, and then we end with the position pair matrix A->B, B->C, C->D, … etc. Now, we can apply the TDLM again to this matrix. This time we are interested in the repeating pattern of the sequence, i.e., how likely will the pairwise sequence, e.g., A->B repeat itself at given time lag.

3 HUMAN REPLAY BUILDS EFFICIENT REPRESENTATIONS

3.1 Introduction

Having developed a method to measure replay in humans I turned to my first question – whether replay re-organizes experience to build a suitable task representation. Having a suitable task representation is crucial for efficient inference and generalization in novel contexts.

Although AI is making impressive strides, humans still learn orders of magnitude faster [94]. Humans are adept at making rich inferences from little data by generalizing structural knowledge from past experience. A crashed car by the roadside, for example, conjures a detailed sequence of past events that were never actually witnessed. It has been theorized that our facility in making correct inferences relies on an internal models of the world, and these are conjectured to be supported by the same neural mechanisms underpinning relational reasoning in space [10,35,37,42,44].

The capacity of replay to play out trajectories [36] and even locations [23] that have never been experienced suggests that replay might be important for building and sampling internal models of space [16,95]. If similar mechanisms do indeed apply in non-spatial scenarios, it would provide a substrate for the powerful inferences and generalization that characterize cognition in humans, whose non-spatial reasoning capacities dwarf those of rodents.

How might replay build efficient representation for inference and generalization? During hippocampal spatial replay events, coherent replay of the same trajectories have been recorded from both medial entorhinal (mEC) [26] and visual cortices [25]. These anatomically distinct regions differ markedly in the nature of their representations. Whilst visual representations encode the sensory properties of a particular event, mEC representations encode structural information (such as spatial relationships), divorced from their sensory properties [59]. One intriguing possibility raised by these observations is that a *factorised* representation may enable prior structural knowledge to constrain the replay of sensory experiences, thereby facilitating novel inference. Such a mechanism could allow incoming sensory events to be replayed in a *new order*, consistent with prior structural knowledge. Approaching drivers, for example, may see first the crashed car, then the road-ice and then the arriving ambulance, but nevertheless replay events the correct order in which these events ensued.

3.2 Results

3.2.1 Unscrambling new objects using a previously learned rule

First, I wanted to test whether replay-like activity is informed by abstract knowledge that is generalized from prior experience. This necessitated a task design wherein learnt sequential structure can be applied to novel sensory stimuli in order to infer a new ordering. To accomplish this, we designed a novel behavioral task, with links to both sequence learning and sensory preconditioning [43,77].

On Day 1, we presented objects sequentially to participants in three stages (Figure 3.1a). In the first stage, participants observed an object sequence Y, Z, Y', Z', with a 300 ms gap between Y and Z and between Y' and Z', and a 900 ms gap between Z and Y'. In the second stage, they observed X, Y, X', Y', with analogous timings. In the third stage, they observed W, X, W', X'. These eight objects actually formed two sequences: WXYZ and W'X'Y'Z'. Before exposure, participants were instructed on a rule that transformed the experienced order into the true underlying order, and after exposure they were quizzed as to the true order.

On Day 2, during MEG scanning, we presented eight new objects, A, B, C, D, A', B', C', D', in a scrambled order which adhered to the same rule learnt in Day 1, and with the same timings across the three stages. We refer to this phase as "Applied Learning". Participants were quizzed on the true order after each run (three runs in total), without feedback. Accuracy at this task stage was 94.44% (vs chance 50%, p < 0.0001, Figure 3.2a), indicating correct application of a previously learned rule to these new objects. After this, participants were shown that the terminal object of one sequence, either D or D', was associated with reward, thereby establishing one sequence as rewarding and one as neutral. We call this phase "Value Learning". Participants were then shown random objects from the sequences and asked whether that object was part of a sequence that would lead to reward. No feedback was provided during these questions, to preclude further learning. Overall accuracy in this phase was 98.55%, indicating correct application of the learned transition model.

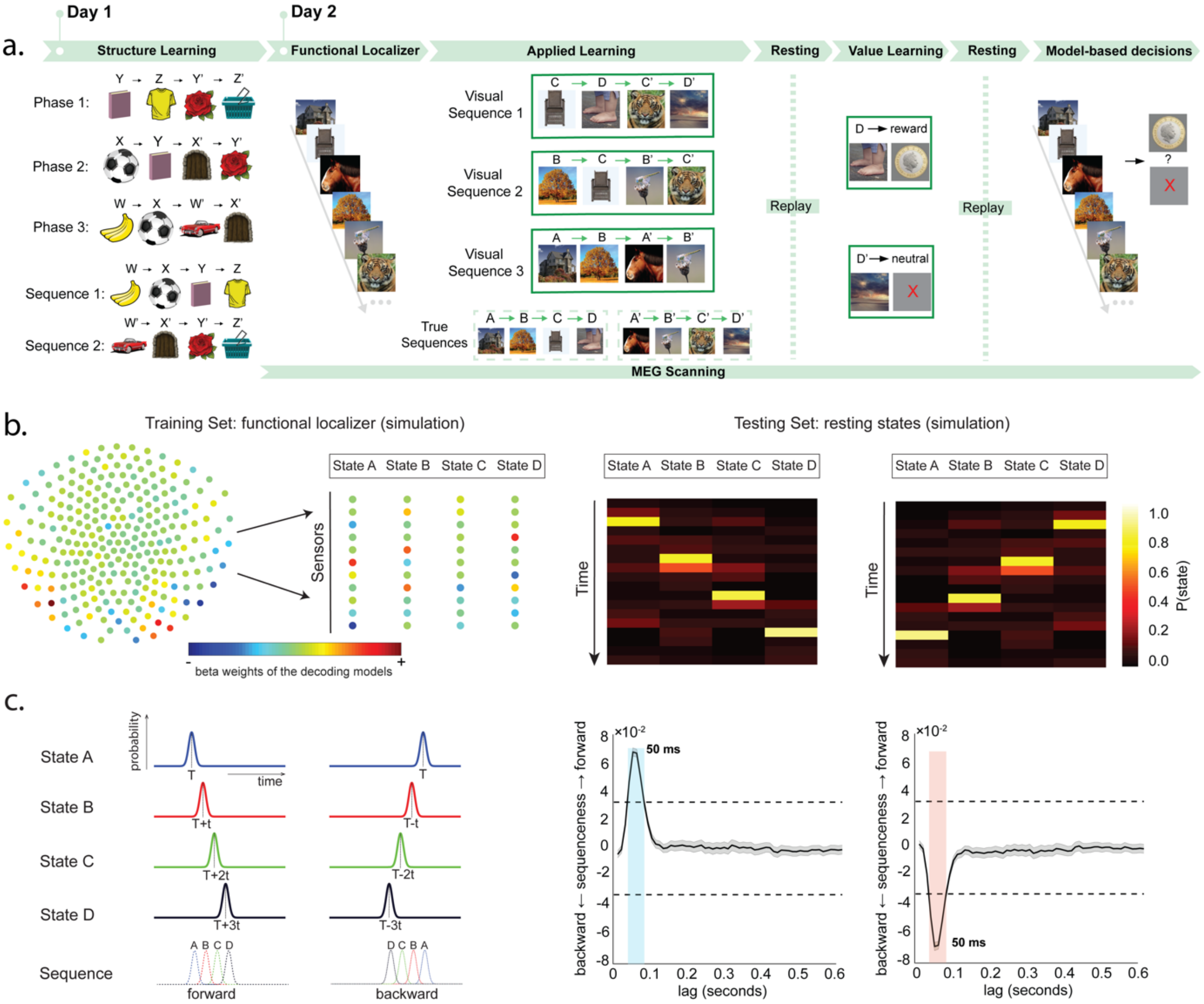


Figure 3.1 Task design of Study 1 and sequenceness measurement

a, Participants were presented with visual stimuli where the correct sequences were scrambled (Study 1). Subjects were pre-trained on Day 1 to re-assemble the stimuli into a correct order. On Day 2, participants underwent a MEG scan while performing a task with the same structure but different stimuli. **b**, Using functional localizer data, a separate decoding model (consisting of a set of weights over sensors) was trained to recognize each stimulus (left). Decoding models were then tested on unlabelled resting data. Examples of forward and reverse sequential stimulus reactivations in simulated data (right). **c,** 'Sequenceness', based on cross-correlation, quantifies the extent to which the representations decoded from MEG systematically follow a transition matrix of interest (left). Evidence for sequenceness (y axis) was quantified at each time lag independently (right), for all possible time lags up to 600 ms (x axis). Dashed line indicates a nonparametric statistical significance threshold (see Methods). Grey area indicates standard error across simulated participants. Coloured areas mark the lags at which the evidence of sequenceness exceeded the permutation threshold in the forward (blue) or reverse (red) direction. All data in this figure are from a simulation where sequences were generated with a state-to-state lag of 50 ms.

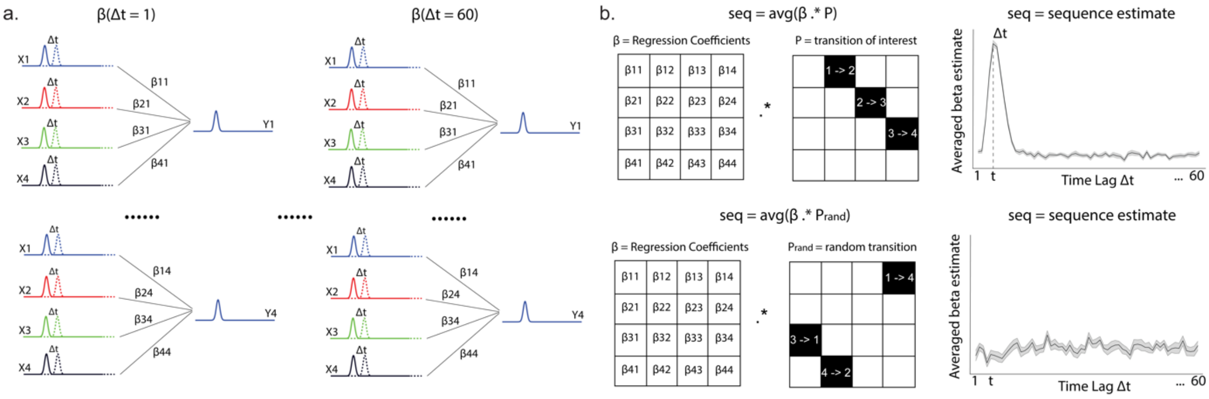


Figure 3.2 Illustration of sequenceness analysis

a, The predictor matrix, X, for this regression is the same matrix, Y, but *time-lagged by Δt* (see inset) to search for linear dependencies between state activations at this time-lag. We constructed separate prediction matrices for each Δt. We then performed time-lagged regression by regressing each lagged predictor matrix X(Δt) onto the state reactivation matrix, Y, resulting in a regression coefficient matrix, with dimension of s states * s states, at each time lag. **b,** This coefficient matrix was then projected onto the hypothesized state transition matrix P, to provide a single measure of sequenceness as a function of time-lag (Δt), and transition structure (P). Evidence of sequenceness for transition of interest (ground truth) vs. random transitions were shown on the top and lower panel respectively. Notably, this regression approach allowed us to include confound regressors in the analysis. We found it helpful to include lagged time-courses at Δt+100ms, Δt+200ms … as confounds to account for 10Hz oscillations that are prevalent during resting activity.

3.2.2 Neural activity spontaneously plays out sequences of new stimuli in an inferred order

Between the Applied Learning and Value Learning phases, there was a resting period of 5 minutes with no task demands or visual inputs. In this resting period, I looked for spontaneous neural sequences that followed either the order of visual experience or an order defined by the previously learned structure. The rest period was intended to be analogous to awake resting periods in rodent studies in which hippocampal replay has been observed in spatial tasks [13,19,96].

To look for structure in the spontaneous brain activity during rest, we needed to be able to decode visual stimuli from patterns of MEG sensor activity (analogous to decoding location from hippocampal cellular activity). Therefore, we included a functional localizer task before the main task. Placing the functional localizer before the main task ensured there was no structure or value information associated with these training data. Here participants simply saw the images that would later be used in the task, presented in a random order, and prior to acquisition of knowledge regarding which visual object played which role in a sequence.

We trained lasso logistic regression classifiers to recognize patterns of brain activity evoked by each visual object. We trained one classifier to recognize each object. Models were cross-validated on training data through a leave-one-out approach to confirm they captured essential object-related features in the MEG signal. We found that the probabilities predicted by each model on held-out data exceeded a significance threshold only when the true stimulus was the same as what that model was trained to detect (Figure 3.3a).

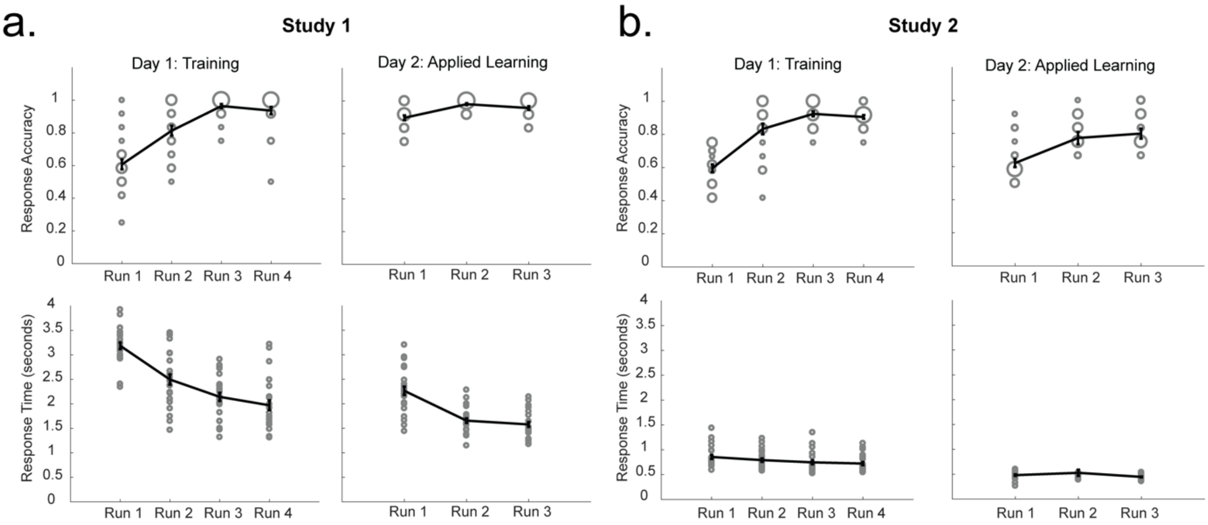


Figure 3.3 Behavioral performance during training on Day 1 and applied learning on Day 2

After learning the structure, participants could quickly unscramble the sequences of different images in both studies. **a,** In Study 1, during training on Day 1, the response accuracy on the sequenceness quiz increased gradually over runs, and the responses time decreased over runs, while on Day 2, during applied learning with different stimuli, most participants understood the correct sequence immediately after first run, and responses were already fast in the first run. **b,** In Study 2, similar effects were found. Note there was a stricter time limit on responses in Study 2 (2 s during training on Day 1, and 600 ms during applied learning on Day 2) compared to Study 1 (5 s during both Day 1 training, and Day 2 learning), which makes the absolute accuracy values not directly comparable. Each circle indicates one unique value. The size of the circle corresponds to the number of participants who have the same value.

We applied trained classifiers to the resting period following the Applied Learning phase. This produced a reactivation probability of each object at each time point during rest (Figure 3.1b). Next, we quantified the degree to which these reactivation probabilities systematically followed particular sequences (Figure 3.1c), using a measure of "sequenceness" we have developed in section 2. This measure defines the extent to which reactivations of object representations follow a consistent sequence defined by a transition matrix.

We considered two transition matrices: the order of visual experience (e.g., C->D->C'->D'), and the true sequences defined by the underlying rule (e.g., A->B->C->D). Significance was tested by randomly shuffling the transition matrix of interest many times and repeating the same analysis, to generate a null distribution of sequenceness. We took the peak of the absolute value of each shuffle across all lags as our test statistic, to correct for the multiple comparisons at multiple lags. For a significance threshold we used the (absolute) maximum of these peaks across time (dashed line in Figure 3.1b).

We found evidence of sequential neural activity that conformed to the rule-defined structure (Figure 3.4b) but not to the visual sequence (Figure 3.4c). These sequences were dominantly in a forward direction. The effect exceeded the permutation threshold from 20-60 ms of state-to-state lag ($p < 1/24 \approx 0.042$, corrected), peaking at 40 ms. Sequenceness appeared within the first minute of rest, remained stable over the next 4 minutes (Figure 3.4c) and was present in the majority of participants (Figure 3.5a).

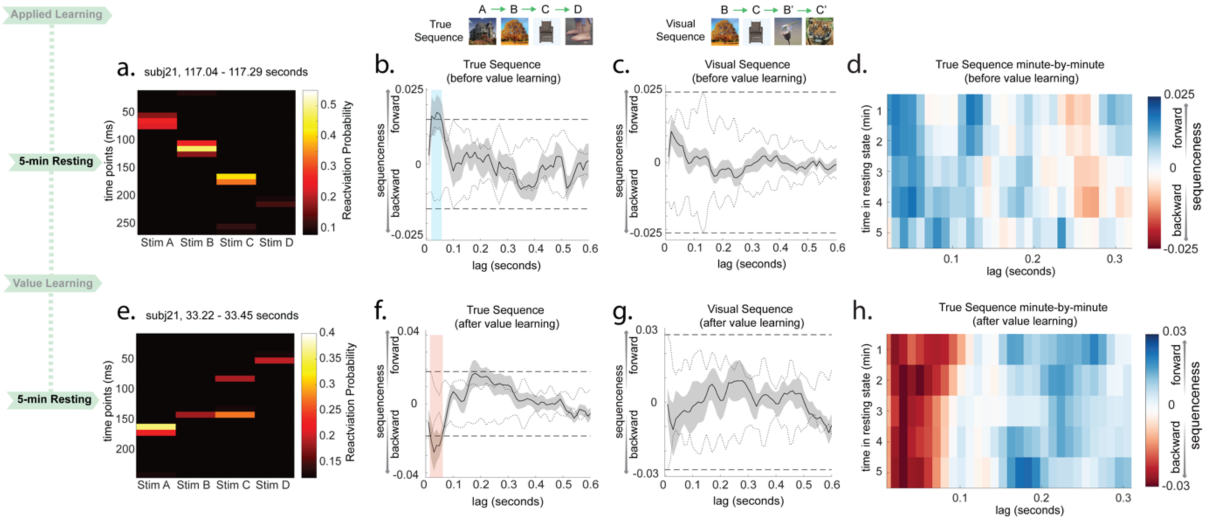


Figure 3.4 Replay follows rule-defined sequence and reverses direction after value learning

In Study 1, examples of sequence events during rests, before and after value learning, are shown from one subject for visualization purposes (**a, e**). Each row depicts reactivation probabilities at a given time point. For statistical purposes, data were analyzed using a ‘sequenceness’ measure (see Methods for details). Stimulus representations decoded from MEG spontaneously played out sequences following a structure defined by the previously learned rule (**b**) and not the visually experienced sequence (**c**). The dotted line is the peak of the absolute value of all shuffled transitions, separately at each time lag; the dashed line is the max across all time lags, which we use as a corrected threshold. During the first rest period, the rule-defined sequences played in a forward direction. **d,** Forward replay of the rule-defined sequence appeared in the first minute of the resting period and remained stable for 4 minutes. **e,** In the second rest period, the rule-defined sequences reversed direction to play in a backwards order. This panel shows an example sequence event. As in the first rest period, there was statistical evidence for replay of the rule-defined sequence (**f**), but not the order of visual experience (**g**). **h,** Reverse replay of the rule-defined sequence after value learning was stable for all 5 minutes of rest. Blue indicates forward sequenceness and red reverse.

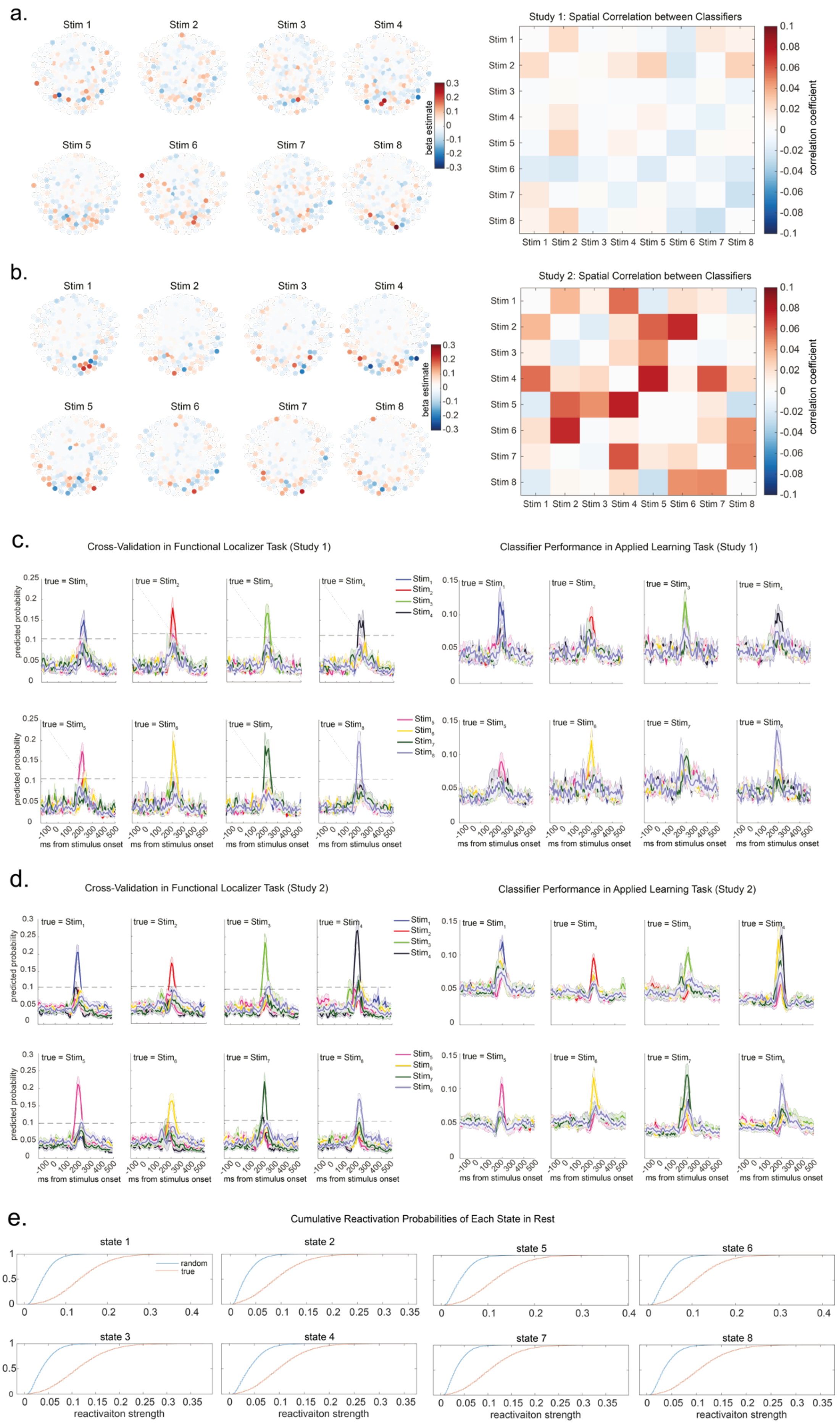

a.
Stim 1
Stim 2
Stim 3
Stim 4
Stim 5
Stim 6
Stim 7
Stim 8
beta estimate
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
Study 1: Spatial Correlation between Classifiers
correlation coefficient
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
b.
Study 2: Spatial Correlation between Classifiers
c.
Cross-Validation in Functional Localizer Task (Study 1)
Classifier Performance in Applied Learning Task (Study 1)
predicted probability
true = Stim$_1$
true = Stim$_2$
true = Stim$_3$
true = Stim$_4$
true = Stim$_5$
true = Stim$_6$
true = Stim$_7$
true = Stim$_8$
Stim$_1$
Stim$_2$
Stim$_3$
Stim$_4$
Stim$_5$
Stim$_6$
Stim$_7$
Stim$_8$
ms from stimulus onset
d.
Cross-Validation in Functional Localizer Task (Study 2)
Classifier Performance in Applied Learning Task (Study 2)
e.
Cumulative Reactivation Probabilities of Each State in Rest
state 1
state 2
state 3
state 4
state 5
state 6
state 7
state 8
random
true
cdf
reactivaiton strength

Figure 3.5 Sensor maps, spatial correlation and classifiers performance of trained Lasso logistic regression models.

a, Sensor map for each state decoding model in Study 1 is shown on the left, with a correlation matrix between classifiers shown on the right. **b,** Sensor maps and correlation matrix is shown for Study 2. **c,** In study 1, leave-one-out cross-validation results for each classifier in functional localizer task is shown on the left. Dotted line indicates the permutation threshold estimated by randomly shuffling the labels and re-doing the decoding process. Classifier performance during applied learning is shown on the right. These plots only use classifiers trained at 200 ms post stimulus onset. The x-axis refers to the timepoint used for testing the classifiers. The curves therefore have a different shape than plots made by varying both the training and testing time. **d,** Study 2 had a very similar pattern. **e,** Cumulative reactivation probability of each state in Study 2 is shown against null distribution (shuffling sensors).

3.2.3 Direction of spontaneous sequences reverses after reward

In rodents, rewards increase the relative frequency of reverse, but not forward replay [20]. Following the Value Learning phase, participants had another 5-min resting period. This second resting period allowed us to test for reward-induced increases in reverse sequences in humans. Consequently, we performed the same sequenceness analysis on the second resting period, after Value Learning. We found the direction of spontaneous neural sequences switched from forward to reverse (Figure 3.4d), exceeding the permutation threshold from 20 to 70 ms of state-to-state lag, and peaking at 40 ms. Again, there was no evidence for sequences corresponding to visual experienced trajectories (Figure 3.4e). The reverse sequenceness effect appeared within the first minute of rest, persisted for all 5 minutes (Figure 3.4f), and was seen for most participants (Figure 3.6). When we examined rewarded and neutral sequences separately, we found evidence that the rewarded sequence alone reversed direction (Figure 3.7a), with the neutral sequence remaining dominantly forward (Figure 3.7b). Interestingly, such reverse replay of rewarding sequences was already present in the Value Learning phase immediately after seeing the rewarding outcome, i.e., money coin (Figure 3.7e). It also continued during online model-based decision-making, where a model of the transitions between stimuli was needed for choice (Figure 3.7f).

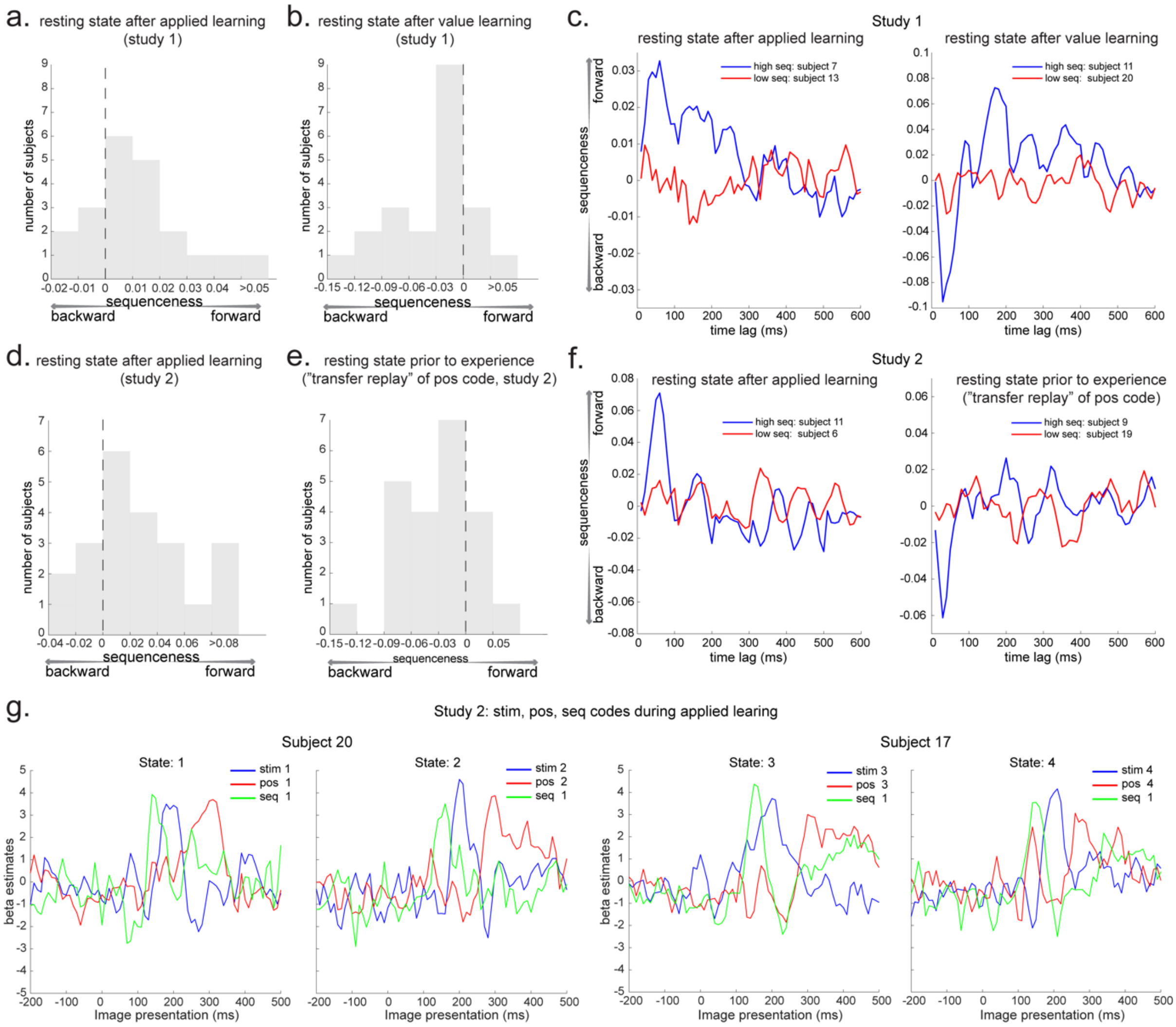


Figure 3.6 Sequenceness distribution across subjects and example data.

a, From Study 1, during a rest period after applied learning, but before value learning, 16/21 subjects had forward sequenceness at 40 ms time lag. **b,** Following value learning in Study 1, 17/21 subjects showed reverse sequenceness. **c,** Sequenceness plot from examples of one "good" subject and "bad" subject in Study 1 are shown both for resting before (left) and after (right) value learning. **d,** From Study 2, forward replay of the true sequence after applied learning was evident in 17/22 subjects. **e,** The position code was played (in reverse direction) prior to experience with the stimuli in 17/22 subjects. **f,** Sequence plot from examples of one "good" subject and "bad" subject in Study 2 are shown both for preplay and replay. **g,** Examples of three codes: *stim*, *pos*, and *seq* codes reactivation from two subjects during applied learning in Study 2 were shown for visualization purpose. These plots (and similar ones for the whole group in figure 5a) show results of a multiple linear regression of the 3 sensor maps associated with the current image (blue), the current position (red) and the current sequence (green) onto the sensor time courses measured after an image is presented during learning. The sensor time courses first represent the sequence, then the image, then the position in the sequence.

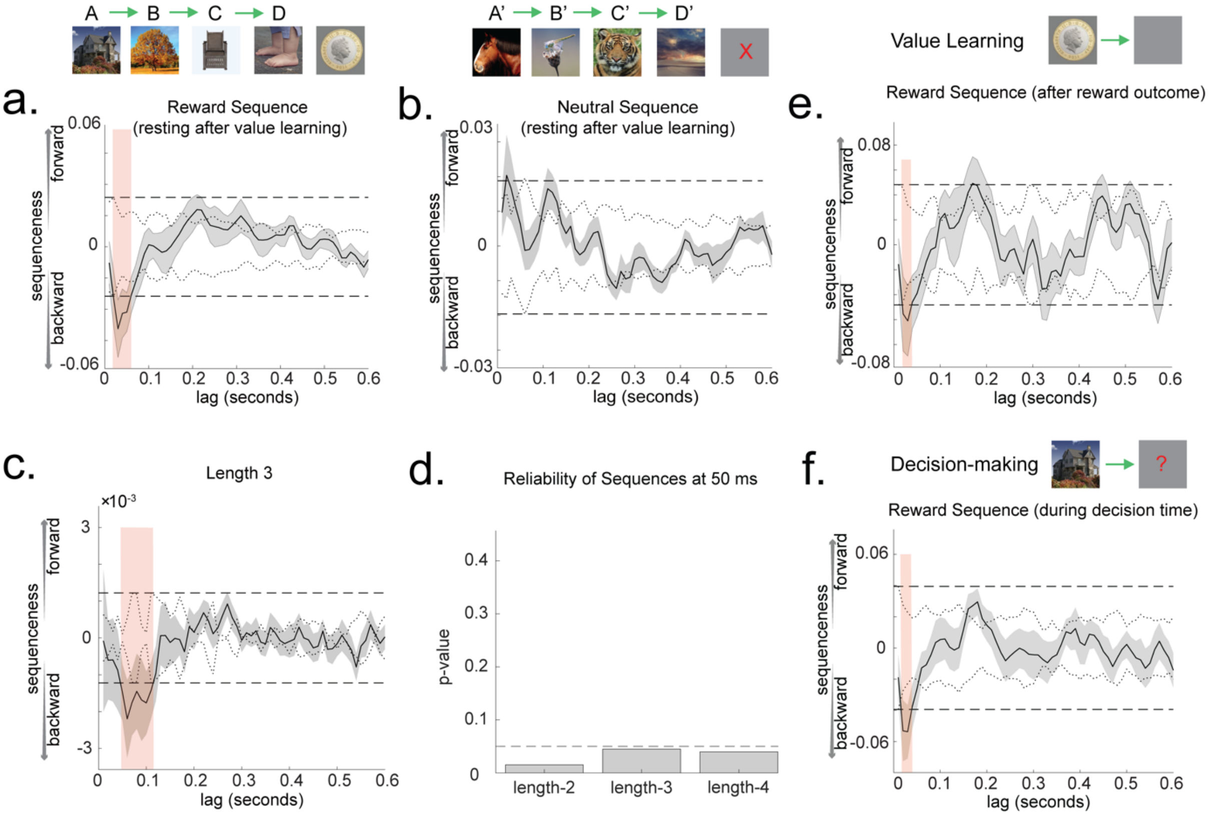


Figure 3.7 Only rewarded sequences reverse direction, and sequences form chains of four objects

a, In value learning, each participant experienced one rewarded sequence and one unrewarded sequence. In the rest period after value learning, the rewarded sequence played backward in spontaneous brain activity. **b,** The unrewarded sequence still trended to playing forward. **c** and **d,** All other panels in the main text show a sequenceness measure that evaluates single-step state-to-state transitions. Here we report a related sequenceness measure that evaluates the *extra* evidence for multi-step sequences, beyond the evidence for single-step transitions (see Methods for details). This measure describes the degree to which, for example, A follows B with the same latency as B follows C. Note, the results here is for reverse sequence, i.e., C->B->A, for example. Sequences of length 3, following the rule-defined order, played out at a state-to-state lag of approximately 50 ms (**c**). At 50 ms lag, there was significant replay of sequences up to the maximum possible length (D->C->B->A). Dashed line at p = 0.05 (**d**). **e**, Replay was not limited to the resting period. Reverse replay of the rewarded sequence began during value learning. **f,** Reverse replay of the rewarded sequence was also evident at the decision phase.

3.2.4 Length-n sequences

Although the transition matrix defined by the order of visual presentation (e.g. C->D->C'->D') differed from the transition matrix defined by the rule (e.g. A->B->C->D), some individual pairwise transitions were common between the two (like C->D). Therefore, we would expect our sequenceness measure to detect some "rule-defined sequenceness" even if the brain replayed only the order of visual presentation. The fact that there was more evidence for the rule-defined sequence than the visually observed sequence renders this interpretation unlikely. However, to rule out this possibility directly we sought evidence for contiguous length-3 or length-4 sequences. We defined a measure of length-n sequenceness that controlled for all lengths up to n-1, measuring the additional evidence for sequences of exactly length-n (see Methods for details).

In the task, there was no overlap between rule-defined sequences (e.g., B->C->D) and visual-order sequences at length-3, meaning length-3 rule-defined sequences could only be reliably observed if neural sequences truly followed a rule-defined ordering. Indeed, we found significant evidence for length-3 reverse replay of rule-defined sequences (Figure 3.7c), peaking at 50 ms state-to-state lag. The additional evidence for length-4 rule-defined sequences was also significant at 50 ms time lag (Figure 3.7d). Together, these data suggest that rapid sequences of non-spatial representations can be observed in humans and have characteristics of hippocampal replay events recorded in rodents. Furthermore, these replay events play out sequences in an order that is implied but never experienced.

3.2.5 Scrambling individual transitions

In Study 1, we found replay-like activity plays out events in an order that is implied, but never actually experienced. However, in that study, participants experienced each individual transition of the implied sequence, albeit not contiguously. Thus, one possibility is that the observed ABCD sequence in brain activity could have arisen from a Hebbian-like encoding of individual transitions. In other words, if B and C were associated, and separately A and B were associated, then ABC could play out through a serial Hebbian associative mechanism. To rule out this hypothesis and determine whether replay truly incorporates abstract structural knowledge, transferred from previous training, I needed a more rigorous test to unambiguously distinguish this hypothesis from a simpler associative mechanism.

Therefore, in Study 2 I designed a task similar to that of Study 1, but with a new jumbling rule where pairwise transitions themselves were disrupted, so that correct sequences could be inferred only using structural knowledge alone (Figure 3.8a). In the first stage, participants observed the sequence, for example, Z', X, Y', Y, with a 500 ms fixation period prior to each stimulus. In the second stage, they observed Z, W', W, X'. These eight objects again formed two true sequences: WXYZ and W'X'Y'Z'. The mapping between presentation order and structural position of the eight objects was randomized across participants. As in Study 1, participants were extensively trained about this structural rule on Day 1.

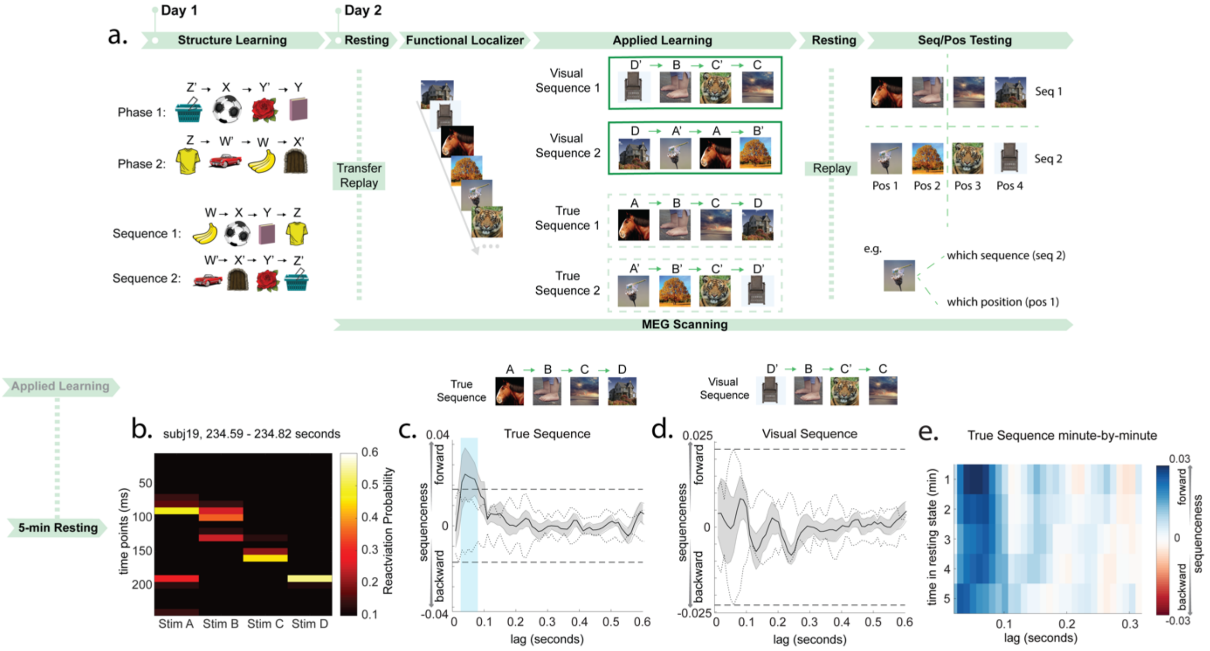


Figure 3.8 Task design of Study 2 and replication of replay following rule-defined sequence

a, In Study 2, not only were sequences scrambled (as was the case in Study 1), but additionally the pair-wise associations were no longer preserved. On Day 1, participants were pre-trained on the rule that defined a re-ordering between the order of stimulus appearance and the task-relevant order. They were also instructed that this same mapping would pertain to novel stimuli on Day 2. On Day 2, participants were shown the novel stimuli while undergoing a MEG scan. **b,** Example of forward replay sequence event from one subject. **c,** During the second rest period there was statistical evidence for forward replay of the rule-defined sequence. **d,** There was no evidence for replay of the visually experienced sequence, as in Study 1. **e,** Forward replay of the rule-defined sequence was stable for all 5 minutes of rest.

Day 2 took place in the MEG scanner, and used the same unjumbling rule as Day 1, but applied now to novel objects. Unlike Study 1, in Study 2 participants were given a 5 minute rest period before any visual exposure to the objects. Participants then performed a functional localizer task where object stimuli were presented in randomized order. Next, these same stimuli were presented in the jumbled order described above. This was followed by another 5-minute resting period. At the very end, participants were shown the stimuli again in a randomized order and were now asked to perform one of two judgments on each stimulus as it appeared. On *position judgment* trials, they were asked to indicate the position (i.e., position 1, position 2, position 3 or position 4) of the stimulus within the sequence it belonged to. On *sequence judgment* trials, they were asked to indicate which sequence (i.e., sequence 1 or sequence 2) the stimulus belonged to. Study 2 had no reward component.

3.2.6 Neural sequences infer a new order, even when pairwise transitions are disrupted

As in Study 1, we again trained a set of classifiers to recognize individual objects from Study 2, using data from the functional localizer. We call these representations the "stimulus code". The validity of these classifiers was again tested using leave-one-out cross-validation. Cross-validated accuracy exceeded a permutation-based threshold for most of the examined post-stimulus epoch, peaking at 200 ms post stimulus onset (Figure 3.5b, Figure 3.9a).

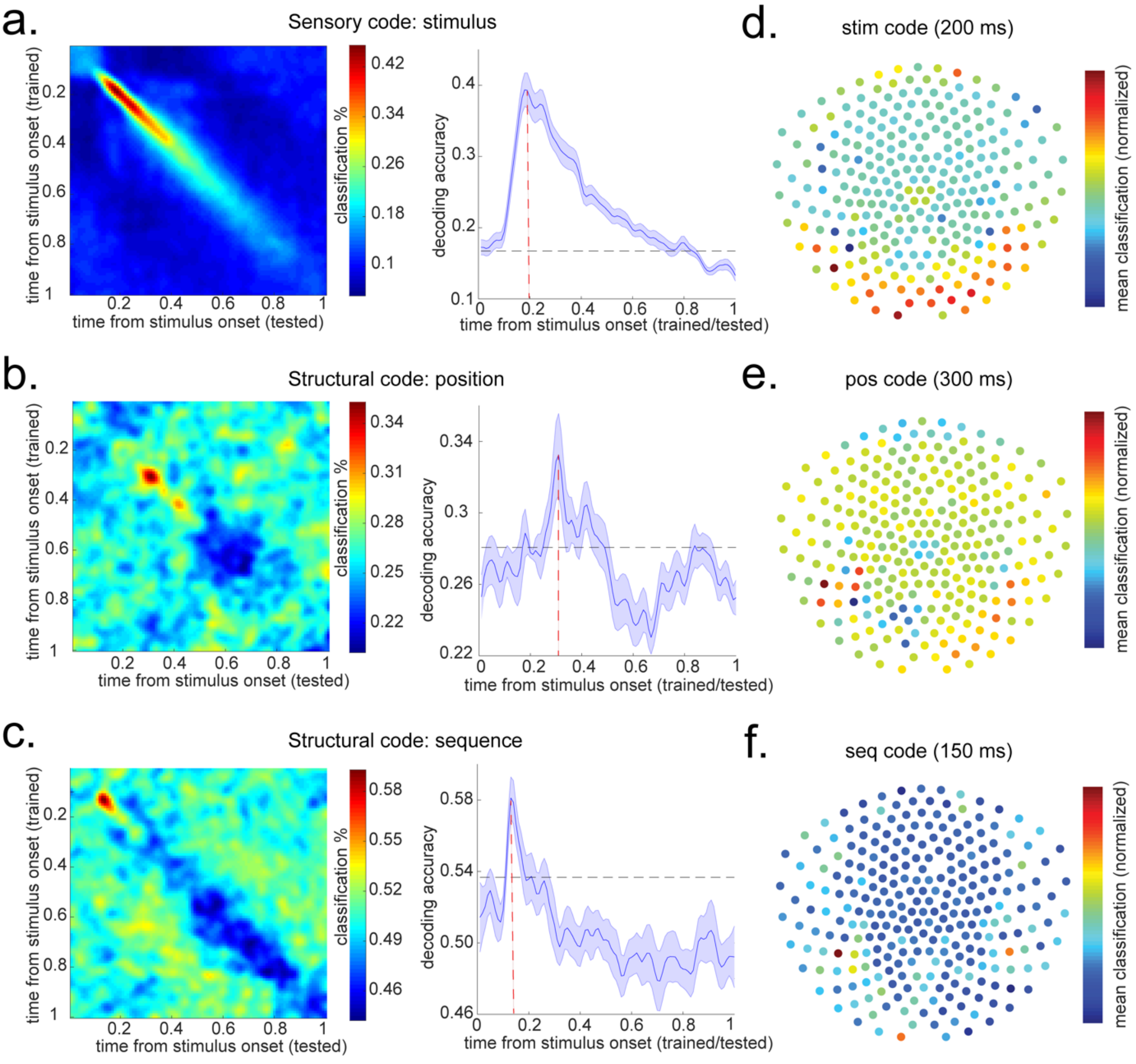


Figure 3.9 Multivariate decoding for sensory and structural representations

a, Using functional localizer data, we trained decoders for the sensory level representation with a leave-one-out cross-validation approach. There was a peak in accuracy at 200 ms post stimulus onset, consistent with previous findings. **b,** To find the peak time point to train position decoders, we trained classifiers on every time point relative to the onset of stimuli, and tested at every time bin relative to the same onsets in the sequence testing block. Each cell of this grid shows cross-validated prediction accuracy on average (left panel). We used the sequence rather than position testing block because of the alignment between the positions and motor responses in the task. Right panel shows the diagonal of the left panel matrix. The peak decoding time was 300 ms after stimuli onset. Dashed lines show 95% of empirical null distribution obtained by shuffling state labels. Shaded area shows standard error of the mean. **c,** Same procedure applied to find the peak time point for sequence identity decoders. This was done on the position testing block to avoid a motor response confound. The peak decoding time was 150 ms after stimulus onset. Structure and sensory codes have distinct encoding in the brain. Averaged decoding accuracy across subjects with each sensor (bootstrapping, n = 2000, with 50 sensors each time) is shown: *stim* decodes individual stimuli (**d**); *pos* decodes order within sequence, invariant to which sequence (**e**); *seq* decodes which sequence, invariant to which stimulus within sequence (**f**).

Using these stimulus code classifiers (trained at 200 ms post stimulus, as in Study 1), we first examined the data from the resting period following the Applied Learning phase. As in Study 1, we found evidence for forward sequenceness following a rule-defined transition matrix (Figure 3.8b), but not the transition matrix of visual experience (Figure 3.8c, d). This effect again peaked at a state-to-state lag of 40 ms (exceeding the permutation threshold from 30 to 70 ms), appeared within the first minute of rest, remained stable for all 5 mins (Figure 3.8e), and existed in the majority of participants (Figure 3.6). Unlike Study 1, however, this stimulus order could not have emerged from simple associative mechanisms, as no correct pairwise associations were present in the visually experienced sequence (Figure 3.8). Therefore, we argue that this reordering implies a transfer of structural knowledge from the previous day.

3.2.7 Neural representations embed structural knowledge in a factorized code

The impact of structural knowledge on replay order raises a question as to whether structural knowledge might itself be a component of the replayed representation. In rodents, entorhinal grid cells replay coherently with hippocampal place cells (e.g., Ólafsdóttir, et al. [26,27], but also see O'Neill, et al. [24]). Unlike place cells, however, grid cells encode position explicitly, with a representation that generalizes across different sensory environments [40]. In our task, objects do not have positions in space, but do have positions within an inferred sequence. Analogously, these positions might be represented explicitly during replay. For example, the 2nd items of each inferred sequence would share part of their replayed representations not shared by the 3rd items. Similarly, we wondered whether items belonging to a particular sequence (e.g., sequence 1), would share representations absent in the other sequence (e.g. sequence 2).

Therefore, in addition to stimulus code classifiers, we trained two additional sets of logistic regression classifiers. One was trained to recognize the position of each stimulus within its respective sequence, regardless of which sequence it belonged to. We call this the "position code". The other was trained to recognize which sequence each stimulus belonged to, regardless of which position it occupied within that sequence. We call this the "sequence code". These classifiers were trained on data from the position judgment and sequence judgment trials, respectively (Figure 3.9, see Methods for details). As an extra precaution against contamination of position and sequence codes by coincidental common sensory features, we regressed out the corresponding stimulus code from each position and sequence classifier (Figure 3.9). We observed that the structural and sensory codes were encoded differently in the brain, with the stimulus code most strongly represented in occipital sensors, while position and sequence codes were reflected more strongly in posterior temporal sensors (Figure 3.9e,f, cf. Hsieh, et al. [97]) .

Next, we asked how representations of structure and sensory information were related. We first used the three sets of classifiers (stimulus code, position code, and sequence code) to probe neural representations during the Applied Learning phase. We found significant activation of all three codes at times closely aligned to their respective training data: the sequence code at 150 ms, the stimulus code at 200 ms, and the position code at 300 ms after stimulus onset (Figure 3.10a). Hence, when a new stimulus appears, neural activity encodes the unique identity of the object, its sequence position and its sequence identity in a factorized representation. This type of representation implies that the same position code is used in different sequences with different objects, providing a potential mechanism for generalization of structural knowledge to support novel inferences.

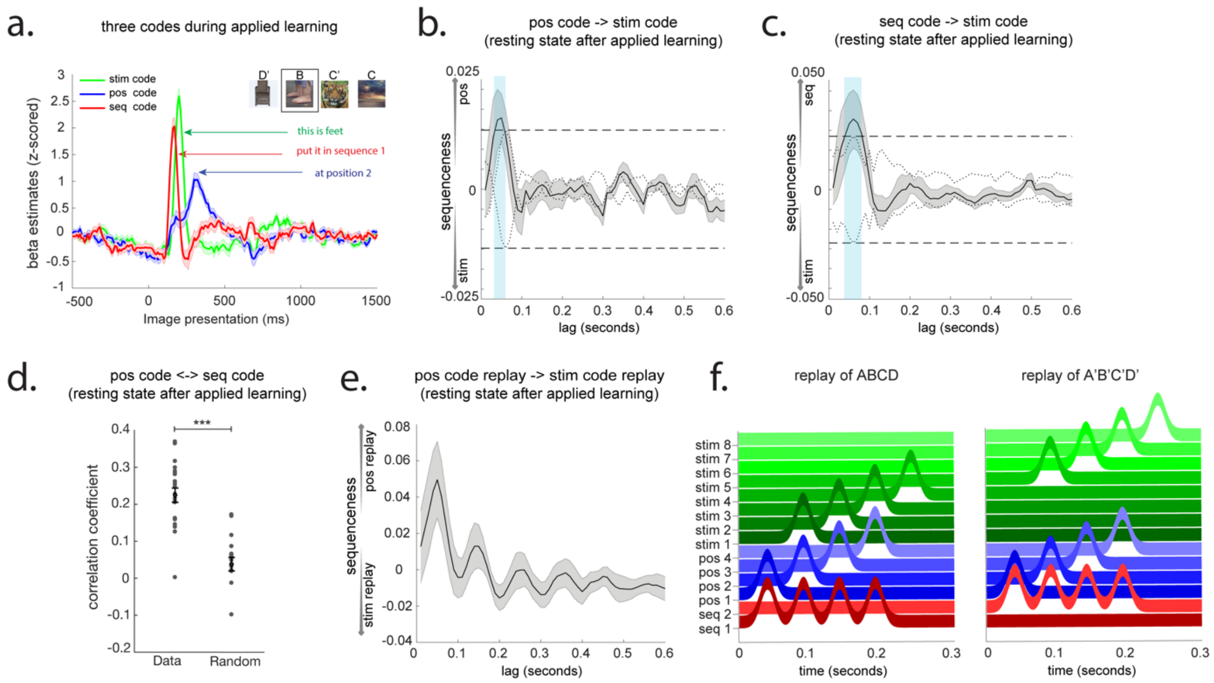


Figure 3.10 Abstract factorized codes play out in synchrony with replay

In addition to decoding models trained to detect representations of individual objects (*stim* code), we also trained additional models to detect representations of "position within sequence" (*pos* code) and "which sequence" (*seq* code). See Methods for details. **a,** During the applied learning phase of Study 2, representations of *seq*, *stim* and *pos* codes for the presented stimulus peaked at distinct times relative to stimulus onset. **b** and **c**, During the second resting period of Study 2, spontaneous reactivation of both *pos* and *seq* codes consistently preceded spontaneous reactivation of the corresponding *stim* code, with ~50 ms lag. **d,** As a validation, we also directly measured the relative timing of *pos* and *seq* activation and found a peak at 0 ms of lag; ***, $p < 0.001$. **e,** Finally, we examined the temporal relationship between replay of stim codes (as shown in Figure 4b) and replay of pos codes (see Methods for details). We found that pos code replay preceded stim code replay events by 50 ms. **f,** Summary of results. During each replay event, stimulus representations (green) were preceded 50 ms by abstract sequence (red) and position (blue) representations.

3.2.8 Abstract representations of sequence identity and position consistently precede object representations during replay events

Are structural representations spontaneously reactivated at rest? To address this, we applied the trained position and sequence code classifiers to the MEG data from the second resting period. As with the stimulus code classifiers, each classifier produced a time series of predicted probabilities. We first examined the temporal relationships between activations of the stimulus, position, and sequence codes. Using the sequenceness analysis described previously, we found that both the position and sequence codes were systematically activated 40-60 ms before the corresponding stimulus code (Figure 3.10b, c). This implies that position and sequence codes were co-activated during rest. Indeed, the zero-lag correlation between unshuffled position and sequence codes were significantly higher than the shuffled correlations (Figure 3.10d; two-tail paired t test, t (20) = 6.47, $p < 0.0001$).

These results show that structural representations consistently precede their corresponding object representations during rest. This raises the possibility that reactivation of a position code could lead an object representation to replay at the correct position within a sequence. To test this idea, we first estimated the replay onset of position codes (e.g., position 1 -> position 2) by multiplying the decoded probability of the first position (e.g., position 1) by the time-shifted probability of the second position (e.g., position 2; see Methods for details). We similarly obtained the time course of the corresponding stimulus code replay. After that, we performed sequenceness analysis on the estimated time courses of position code replay and stimulus code replay: asking whether replay of position codes has a temporal relationship to replay of corresponding stimulus codes. We found that replay of position code temporally led replay of stimulus code, with a peak at 50 ms of lag (non-parametric one-sample Wilcoxon sign rank test against zero, $p = 0.013$, Figure 3.10e). These results are consistent with a model outlined in Figure 3.10f, where each individual object representation in a replay event is preceded by both sequence and position representations. We speculate that these abstract structural representations contribute to retrieving the correct object for the current place in a sequence.

3.2.9 Abstract position representations play in sequences prior to new object experience ('transfer replay')

In rodents, prior to experience with a novel environment, hippocampal place cells play out spontaneous trajectories, which later map onto real spatial trajectories when an environment is experienced [18]. This has been called 'preplay' and is proposed to encode general information about the structure of space. Our task allowed us to ask a similar question: whether abstract representations of task structure, defined using Day 2 objects, are played before those objects are ever seen.

For this analysis we took advantage of the first resting period at the beginning of the MEG scan, before participants experienced Day 2 objects. Using trained models of position codes, we performed the same sequenceness analysis as previously described, using the transition matrix position 1 -> position 2 -> position 3 -> position 4. Examples of sequential reactivations of position codes during the first resting period are shown in Figure 3.11a. Statistically, we found significant reverse sequenceness, peaked at a 30 ms position-to-position lag (n = 21, $\beta = -0.036 \pm 0.008$), exceeding a permutation threshold from 20 ms to 60 ms time lag (Figure 3.11b). We refer to this phenomenon as 'transfer replay', because it links Day 2 objects to previous experience, and to avoid confusion with the complex preplay literature.

Transfer replay appeared within the first minute of rest (Wilcoxon signed-rank test, $p = 0.011$), remained stable for all 5 mins of rest (2nd minute, $p = 0.011$; 3rd minute, $p = 0.0006$, 4th minute, $p = 0.007$, 5th minute, $p = 0.0005$, Figure 3.11c), and was evident for most participants (Figure 3.6e; examples of individual sequences in Figure 3.6f, right). As a sanity check, we also tested for replay of stimulus codes during the first resting period, which should not be possible since stimuli had not yet been experienced. Reassuringly, we found no evidence for such sequenceness (Figure 3.11d).

If transfer replay constitutes a 'rehearsal' of structural knowledge, we might ask whether individuals with transfer replay are quicker to apply structural knowledge to new objects. Consequently, we measured the relationship between transfer replay strength and position code reactivation during each run of learning. Participants with greater transfer replay had less overall position code reactivation, and this effect was driven by a decrease in position code reactivation over learning in participants with high, but not those with low, transfer replay ($p = 0.007$ interaction term in regression; Figure 3.11e). We speculate that this reflects individuals with high transfer replay rapidly learning a position-to-stimulus mapping

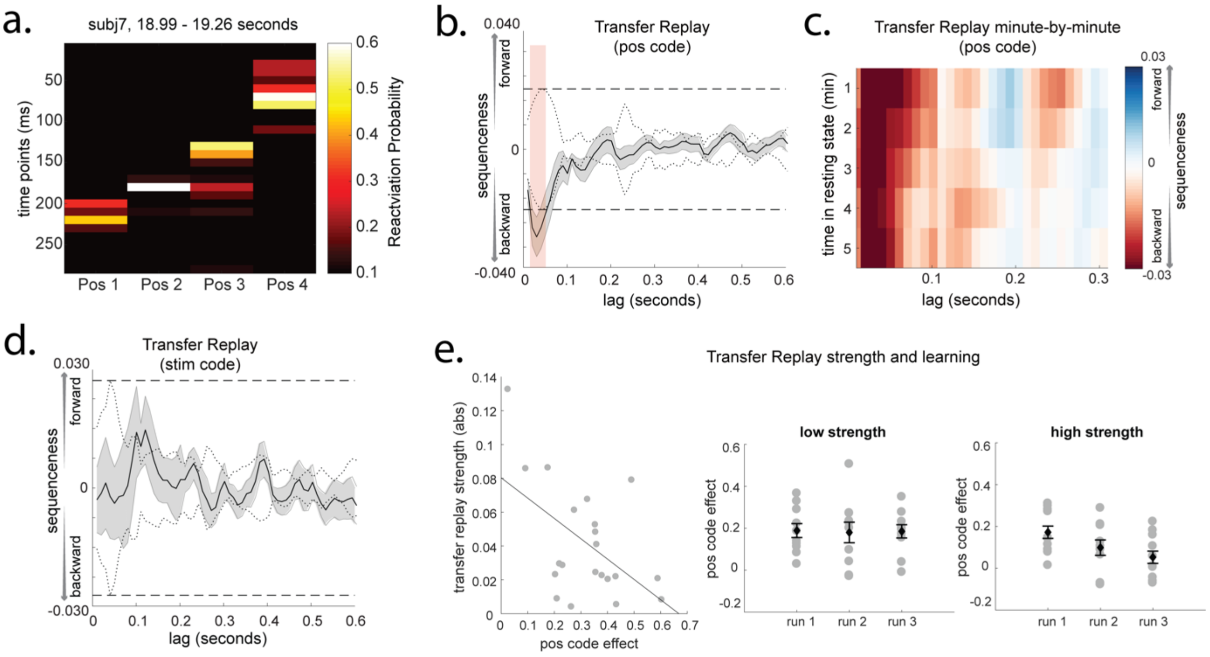


Figure 3.11 'Transfer replay' of position code before exposure to new stimuli

On Day 2, participants had a rest period in the scanner before exposure to the novel objects. During this rest period, we observed reverse sequences of *pos* code reactivations (i.e., position 4, position 3, position 2, position 1). **a,** Example of a reverse sequence event made up of *pos* codes. **b,** Statistically, there was strong evidence across subjects for reverse sequences of *pos* codes. **c,** This reverse sequenceness was stable for all 5 minutes of rest. **d,** As a sanity check, we also looked for replay of the *stim* code in the first rest period, which should not be possible because the stimuli have not yet been seen. We found no evidence of such activity. **e,** Between subjects, the strength of positional preplay (from panel a of this Figure) was negatively correlated with the degree of *pos* code activation during applied learning (from Figure 5a, red trace). Subjects with low preplay (by median split) expressed the position code more strongly throughout learning. Conversely, subjects with high preplay had a steep falloff in position code activation across learning. Each grey dot indicates an individual subject, with error bars representing standard error of the mean across subjects.

3.2.10 Power increase in sharp-wave ripple frequencies around replay events

In rodents, spontaneous offline replay events co-occur with bursts of high frequency (120-200 Hz) local field potential power known as sharp wave ripples (SWRs) [98]. To see if we could detect a similar phenomenon in humans, we performed time frequency analysis in both studies (Figure 3.12a, see also Figure 3.13a, b for time frequency analysis in longer epoch and inter-replay-intervals from both studies). We evaluated frequencies up to 150 Hz, the maximum allowed by our data acquisition methods.

Individual replay events were defined as moments with high probability of a stimulus reactivation that were also followed 40 ms later by high probability of reactivation of the next stimulus in the sequence (see Methods for details). At the onset of replay, we found a power increase at 120 - 150 Hz, compared to a baseline period 100 to 50 ms before replay events. This increase lasted approximately 100ms and was significant in both studies (Figure 3.12a; cluster-based permutation test with cluster forming threshold $t > 3.1$ and number of permutations = 5000).

To examine this effect in sensor space, we averaged across power changes in the frequency range 120-150 Hz and then ran permutation-based analysis on the sensors*time map (cluster forming threshold $t > 3.1$, 5000 permutations). The pattern of sensors with increased SWR-band power at replay onset was similar in both studies (Figure 3.12b) and strongly resembled the distribution over sensors that corresponds to intracranially recorded hippocampal LFP [99].

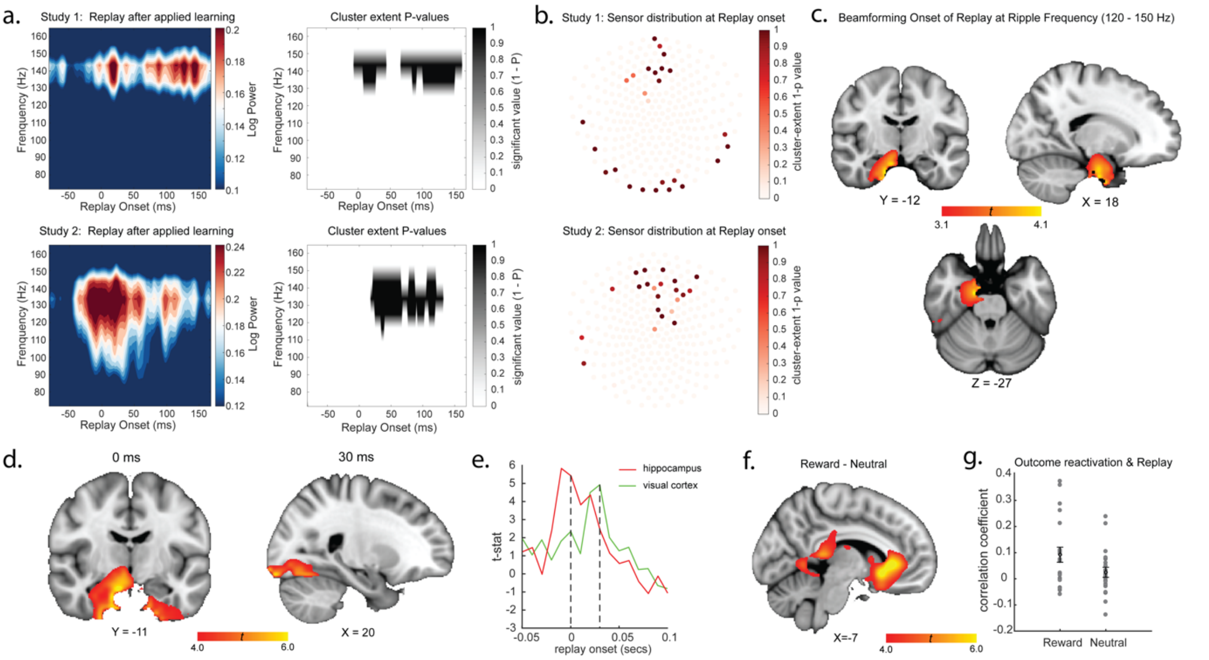


Figure 3.12 Replay coincides with ripple-band power, which source localizes around hippocampus

a, Left, In the ripple frequency band (120-150 Hz), there was a significant power increase (averaged across all sensors) at the onset of replay, compared to the pre-replay baseline (100 to 50 ms before replay). Right, a cluster-based permutation test (cluster forming threshold t > 3.1, number of permutations = 5000) identified a significant cluster around 140 Hz. This effect replicated across Study 1 and Study 2. Note, replay events were excluded if there was another replay event in the baseline period (see figure S7). **b,** Sensor distribution of p values (plotted as 1-p) for ripple-band power increase at replay onset. We found similar sensor patterns in Study 1 and Study 2. **c,** combining data from two studies, source localization of ripple-band power at replay onset revealed significant hippocampal activation (peak MNI coordinate: X = 18, Y = -12, Z = -27). **d,** We also contrasted broadband power at replay onset times against the pre-replay baseline. This contrast also found activity in medial temporal lobe that encompassed bilateral hippocampus (peak MNI coordinate: X = 16, Y = -11, Z = -21). When performing the same contrast but using onsets 30 ms after replay, we found visual cortex (peak MNI coordinate: X = 20, Y = -97, Z = -13). Source image thresholded at *t* > 4.0 (uncorrected) for display purposes. Both the hippocampus (at 0 ms) and visual cortex (at 30 ms) survived whole-brain multiple comparison correction based on a non-parametric permutation test. **e,** The time course of hippocampus at its peak MNI coordinate is shown in red, while visual cortex at its peak MNI coordinate is shown in green. **f,** We also examined the contrast of rewarded sequence replay onset against unrewarded sequence replay onset, in Study 1. In this contrast we found vmPFC activation. **g,** We trained decoding models to detect representations of reward outcomes and neutral outcomes. Spontaneous reactivation of reward outcome representations was correlated with replay of the rewarded sequence. There was no correlation between reactivation of the neutral outcome and replay of the neutral sequence.

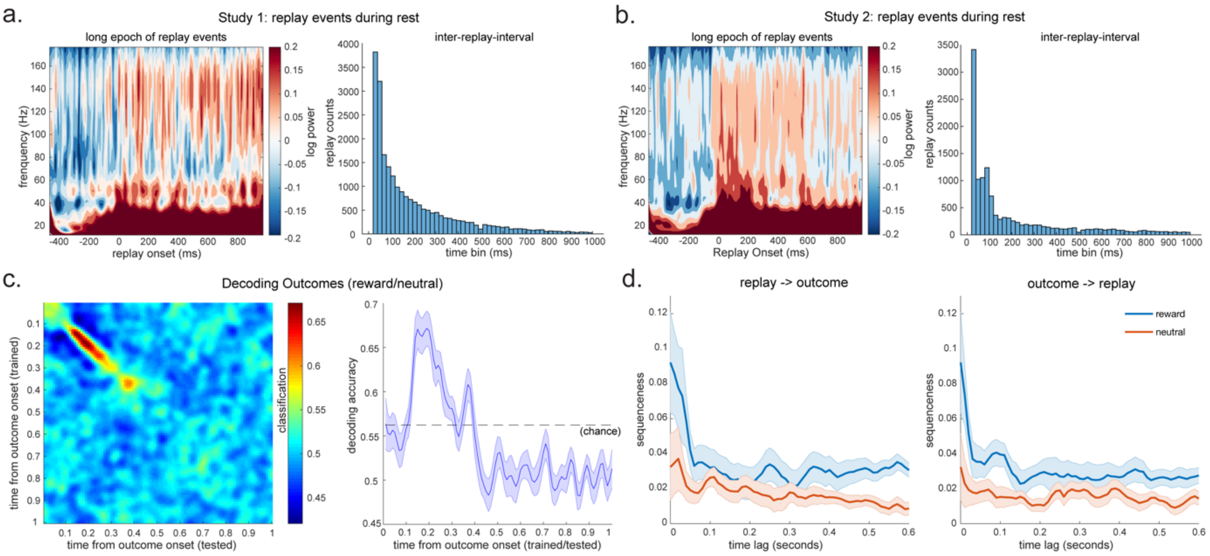


Figure 3.13 Replay events in long epoch, inter-replay-interval, and outcome reactivation

a, b, Time-frequency maps shown for an extended epoch after replay events in study 1(a) and 2 (b). 0ms is the onset of replay events. Notably these are only replay events which are not preceded by other replay events in the previous 100ms. Note that the increase in ripple-band and low frequency power extends for several hundred milliseconds. This can be explained by the fact that replay events occur in clusters. Histograms show the inter-replay intervals across all replay events in all subjects in study 1(a) and study 2(b). The modal replay onset time is immediately following the previous replay event. The heavy tail of the distribution indicates that there are also periods with no replay events. **c,** To find the peak time point to train outcome decoders in the value learning phase from Study 1, we trained classifiers on every time point relative to the onset of outcome, and tested at every time bin relative to the same onsets. Each cell of this grid shows cross-validated (leave-one-out) prediction accuracy. We found the peak decoding accuracy was around 200 ms after the stimuli onset. **d,** The reward outcome reactivated at the same time as the onset of replay of the rewarded sequence during rest, while no such relationship was observed for neutral outcome and replay of neutral sequence.

3.2.11 Source localization of replay

In general, source localizing neural activity measured using MEG is treated with caution due to inaccuracies in the forward model and the priors used to invert it [100]. With this caveat in mind, we asked about the likely anatomical sources of replay using resting state data combined from both studies.

We epoched the data using replay onset events and beamformed power in different frequency bands into source space (see Methods for details). When we considered either ripple frequencies alone (120-150Hz - Figure 3.12c) or broadband power across all frequencies (0-150Hz - Figure 3.12d left), power increases at the onset of replay events were associated with sources in medial temporal lobe. Notably, broadband power increases 30ms *after* replay events were associated with sources in visual cortical regions (Figure 3.12d right). Each of these power increases survived statistical thresholding by permutation test (p<0.05 corrected; see Methods). For display purposes, we extracted the peak coordinate from hippocampus and visual cortex respectively and plotted the respective time courses of their broadband power (Figure 3.12e). In future experiments it will be intriguing to test the idea that relational knowledge embedded in medial temporal areas orchestrates replay of sensory representations.

To localize the neural difference between replay of rewarded and neutral sequences in Study 1, we contrasted the onset of the rewarding sequence against the onset of the neutral sequence. We found activation in ventral medial prefrontal cortex (vmPFC), extending to the ventral striatum (Figure 3.12f). Given vmPFC is known to encode value, we tested whether reward representations were associated with the rewarded sequence during rest. As a supplemental analysis, we separately trained a classifier, using data from the times of outcome deliveries, to distinguish reward from non-reward outcomes (Figure 3.13c). We then applied this classifier to the second resting period in Study 1. We found that reward outcome representations were coincident with the onset of a rewarded sequence (two-tail one sample t test against zero, t (20) = 3.20, p = 0.005), with no such relationship between neutral outcome representation and onset of the neutral sequence (t (20) = 1.17, p = 0.26) (Figure 3.12g, Figure 3.13d). We speculate that a reward representation in vmPFC may play a role in initiating replay of a rewarded sequence.

3.3 Discussion

At rest, the hippocampal-entorhinal system and neocortex spontaneously play out rapid-fire sequences of real and fictive spatial trajectories, decoupled from current sensory inputs [16]. Here, I provided evidence that such replay exists in non-spatial domains and can be measured non-invasively in humans. I show that fictive replay does not merely stitch together experienced sub-trajectories, but it also constructs entirely novel sequences in an inferred order determined by abstract structural knowledge. Finally, we observe that each replayed sequence is comprised not only of representations of the stimuli in the sequence, but also of representations of an abstract task structure. We propose that this abstract replay is a mechanism for generalizing structural knowledge to new experiences.

3.3.1 Non-spatial replay

Spatial replay is a remarkable neural mechanism that bridges between our understanding of circuit mechanism and computational function [16,42]. If replay is a ubiquitous feature of brain activity that extends beyond the spatial domain, it may contribute to learning and inference that relies on arbitrary conceptual knowledge. Rapid spontaneous sequences of non-spatial representations have previously been observed in humans [53]. Three features of the current study bring these measurements closer to rodent hippocampal replay – the sequences are present during rest [13], they reverse direction after rewards [20], and they coincide with an increase in source-localized hippocampal power in ripple frequency [98]. Together with the degree of time-compression, these findings provide strong parallels and convergence with rodent replay events seen during sharp-wave ripples.

3.3.2 Factorization and replay of inferred sequences

In rodents, there is evidence for replay of never-before-experienced sequences that are consistent with the geometry of a spatial map [23,36]. Here we ask whether, like spatial geometry, learnt arbitrary structures impose constraints on replay. Participants first learned an arbitrary unscrambling rule that defined a sequence over a set of objects. After experiencing a novel set of objects, replay immediately sequenced the objects according to this rule rather than the order of experience. This can be viewed as "meta-learning" [101-103], insomuch as previous learning facilitated rapid integration of new knowledge.

Generalization of learned structures to new experiences may be facilitated by representing structural information in a format that is independent from its sensory consequences, as do grid cells in spatial experiments. Factorized representations are powerful as they allow components to be recombined in many more ways than were actually experienced [10,104,105]. The ability to recombine knowledge during replay may allow disparate observations to be compiled into meaningful episodes online [95,106]. Equally important, it may allow simulated experiences to drive cortical learning offline. Using time to bind together new combinations [107], is attractive as it avoids the requirement for a combinatorically large representational space (see also Kumaran and McClelland [108]).

In rodents, disruption of replay-containing sharp wave ripple events degrades subsequent spatial memory performance [109,110], implying a causal role for spatial replay. However, we cannot be certain the same causal role exists for the factorized replay described here. Future experiments using online detection and disruption of replay events will be needed.

3.3.3 Anatomy of replay

Although it is difficult to make confident anatomical statements with MEG, an advantage is simultaneous recording from the whole brain. Given the weight maps of our object classifiers, it is most likely that the spontaneous sequences we detected were sequences of neocortical representations. At the same time as these sequences appeared, there was also a transient increase in power around 140 Hz which source localized to the medial temporal lobe. These observations are consistent with the idea that replay may be coordinated between hippocampus and neocortical areas [25]. Moreover, by taking whole brain measurements, we were able to make an observation that has not been reported in rodents. vmPFC activated 10 ms prior to reward-associated reverse replay events, hinting that such events might be instructed by the reward system. These results commend simultaneous recordings in rodent prelimbic cortex and hippocampus during rest.

3.3.4 Transfer replay

On Day 2, abstract representations of position replayed after learning and also played out before applied learning (in the first rest period). This raises an interesting analogy to what is termed "preplay" in rodent research [18]. In our human data, these same position codes were later tied to new stimuli during the learning phase and during post-learning replay on Day 2. Furthermore, the degree to which they played out before learning predicted the effects seen during applied learning. Therefore, it is plausible that transfer replay is used to support learning about new stimuli. In this spirit, transfer replay bears resemblance to preplay reported in rodent hippocampus. Indeed, it is also possible that preplay sequences reported in rodent hippocampus may have been learnt through experience of other sequences. However, we cannot determine if the pre-experience sequences in our data indeed reflect learnt structure or a preconfigured neural dynamic as previously suggested in the literature [56,71]. An interesting possibility is the position codes might not only be abstracted over different sets of objects within this particular task, but in fact be common to other transitively ordered sequences [111]. In rodents, preplay has been suggested to reflect knowledge of a common structure of space [18,112]. Similarly, tasks which involve inference on linear structures may benefit from representations of their shared underlying statistics [10,113].

3.4 Conclusion

The ability to measure replay in humans opens new opportunities to investigate its organization across the whole brain. Our data suggest powerful computational efficiencies that may facilitate inferences and generalizations in a broad array of cognitive tasks.

3.5 Methods

3.5.1 Participants

Twenty-five participants (aged 19-34, mean age 24.89) participated in the first study. Eleven were male, and one was left-handed. Three were excluded before the start of analysis because of large movement (>20 mm) or myographic artefacts. Data from another participant was unusable due to missing triggers, leaving 21 participants in total for further analyses in Study 1. A separate cohort of twenty-six participants (aged 19-34, mean age 25.48) participated in the second study. Ten were male and two were left-handed. Two of these were excluded before the start of analysis due to movement-related (>20 mm) or myographic artefacts. Another two participants were excluded due to failure of understanding of the task, leaving 22 participants in total for further analyses in Study 2. All participants were extensively trained the day before the MEG task and had a mean accuracy of at least 80% on probe trials of the sequence representation.

All participants were recruited from the UCL Institute of Cognitive Neuroscience subject pool, had normal or corrected-to-normal vision, no history of psychiatric or neurological disorders, and had provided written informed consent prior to start of the experiment, which was approved by the Research Ethics Committee at University College London (UK), under ethics number 9929/002.

3.5.2 Task

In the Study 1, we exploited a revised sensory preconditioning paradigm (Figure 3.1a). There were eight visually distinct objects, each representing a different state in a sequence. Four objects constituted one single sequence, providing for two distinct sequences (i.e., A->B->C->D; A'->B'->C'->D'). Participants were initially presented with objects shuffled in order (e.g., C->D; B->C; A->B); and were subsequently required to rearrange them in a correct sequential order (e.g., A->B->C->D) without ever having experienced this full trajectory. Participants were trained a day before scanning and with different stimuli, meaning they were trained on the actual structure of the task. Then, on a second day, participants underwent MEG scanning doing a similar task but now performed with different stimuli (see Appendix I for standardized instructions in Study 1). The task was implemented in MATLAB (MathWorks) using Cogent (Wellcome Centre for Human Neuroimaging, University College London).

In the MEG scanner, participants first completed a "functional localizer" task. This task was designed to elicit neural representations of the stimuli, and these were subsequently used to train classification models. In brief, the name of a visual object appeared in text for a variable duration of 1.5 to 3 s, followed immediately by the visual object itself. On 20% of trials, the object was upside-down. To maintain attention, participants were instructed to press one button if the object was correct-side-up, and a different button if it was upside-down. Once the participant pressed a button, the object was replaced with a green fixation cross if the response was correct and a red cross if the response was incorrect. This was followed by a variable length inter-trial interval (ITI) of 700 to 1,700 ms. There were two sessions, each session included 120 trials, with 24 correct side-up presentations of each visual object in total. Only correct-side-up presentations were used for classifier training. The trial order was randomized for each participant and visual object and state mapping was randomized across participants.

Participants were then presented with the stimuli and were required to unjumble the "visual sequence" in their minds into a correct order, i.e., the "structure sequence". There were three phases in each block, where each phase comprised two pairwise associations, one from each structure sequence. In each phase, objects from the two associations were presented consecutively (i.e., the "visual sequence"), each stimulus was presented for 900 ms, followed by an inter-stimulus interval (ISI) of 900 ms, then followed by the other pairwise association. Each phase was repeated three times, then followed by the next phase. There were three blocks in total, each block was followed by multiple choice questions designed to probe whether participants had learnt the correct sequence (i.e., the "structure sequence"). At each probe trial, the probe stimulus appeared for 5 s during which participants need to think about which object followed the probe stimulus in the structure sequence, and then selected the correct stimulus from two alternatives. No feedback was provided. There was a 33% possibility that the wrong answer came from the same sequence but was preceding instead of following the probe stimuli. This setup was designed to encourage participants to form sequential rather than clustering representations (i.e., which sequence does this object belong to).

After the Applied Learning, participants had a 5 mins rest period, during which they were not required to perform any task. After the 5 min rest period, participants were then taught that the end of one sequence led to reward of money, while the end of the other did not, in a deterministic way. In each trial, participants saw the object of each end of the sequence (i.e., D or D') for 900 ms, followed by an ISI of 3 s, and then either received a reward (image of a one-pound sterling coin) or no-reward (blue square) outcome for 2 s, followed by an ITI of 3 s. Objects appeared 9 times, for a total of 18 trials. Participants were required to press one button for the reward, and a different button for non-reward. Pressing the correct button to 'pick up' the coin led to a payout of this money at the end of the experiment (divided by a constant factor of ten), and participants were pre-informed of this. After value learning, participants had another rest period, for 5 mins, without any task demands.

As a final assignment, participants were asked to perform a model-based decision-making task. Here they were asked to determine whether presented stimuli led to reward or not. In each trial, an object was presented on the screen for 2 s, and participants were required to make their choice within this 2 s time window, followed by ITI of 3 s. Each stimulus was repeated 5 times such that there were 40 trials in total, 20 for each sequence. The trial order was fully randomized with a constraint that the same stimulus would not appear consecutively. No feedback was provided after a response so as to eliminate learning at this stage. After the task, participants were required to write down two sequences in the correct order. All participants were 100% correct, suggesting they maintained a task structure representation until to the end of the task.

In the Study 2, not only were sequences jumbled but in addition the pairwise associations were also disrupted (Figure 3.8a). We aimed to 1) replicate what we had found in Study 1, and if so, ask whether replay follows the structure of a sequence without temporal proximity 2) determine whether the structure representation was encoded explicitly, and establish its role in learning and replay.

We separated neural representations at sensory and structure level and investigated how structure representation guided replay of sensory information into correct sequences (Figure 3.8a). On Day 1, each participant was given a template for a mapping between visual presentation order of stimuli and their positions in the structure sequences. Participants were told this mapping remained the same on the second day of MEG experiment, with the sole difference being the stimuli (i.e., a different set of 8 stimuli were used on Day 2). The visual order of stimuli was now completely jumbled up, not only in terms of the order of sequences, but also the relationship of pairwise associations was disrupted in visual presentation (see Appendix II for standardized instructions in Study 2). This meant that participants had to use structure knowledge to explicitly figure out correct sequences. The mapping between visual order and structure order was randomized across participants.

On Day 2, participants were required to perform a MEG experiment with a different set of stimuli, but under the same structure. First, participants were allowed a 5-min rest period at the beginning of MEG experiment before any experience of the new stimuli. This was similar to the setting of Dragoi and Tonegawa [18], where they found a "preplay" sequence depicting future spatial trajectories in the absence of prior experience. If the knowledge of relational structure is explicitly encoded, we predicted we would observe preplay depicting this structure representation in sequence.

After a resting period, participants performed a functional localizer task, as in Study 1. This was then used to train classification models of stimuli. Note, the functional localizer task is before Applied Learning, such that no structure information is associated with these training data. These decoding models therefore capture a sensory level in neural representations of stimuli (i.e., stimulus code).

Participants were then presented with the stimuli and required to unjumble the "visual sequence" into a correct order, i.e., the "structure sequence" based on the template they learnt the day before. There were two phases in each block, each phase comprised four images with each stimulus presented for 900 ms, followed by an inter-stimulus interval (ISI) of 900 ms. Each phase was repeated three times, then followed by the next phase. There were three blocks in total, each block followed by multiple choice questions without feedback, similar to the first study. However, differing from the first study, the probe stimulus appeared alone (without showing the alternative images) for 5 s, then the two alternatives were shown for 600 ms during which participants needed to make a response. This manipulation was to limit further any potential associative learning when the stimuli were presented together.

After the Applied Learning, participants were given a rest for 5 mins again. We were interested 1) to replicate our finding in the first study, i.e., replay stitched together independent events into a sequence that is constrained by the relational structure; 2) to understand how structure representations guide sensory information into a replay of correct sequences.

Finally, participants were required to determine the corresponding position or the sequence identity of that stimulus in its structure constrained sequence. This testing aimed to elicit the neural representation of structure information associated with each stimulus. There were two blocks, one block was for position testing, and the other was for the sequence identity testing. The order of the two blocks was counterbalanced across participants.

In each trial, object was presented on screen for 2 s, and participants were required to determine either its associated position or the sequence identity within the 2 s time window, followed by ITI of 3 s. Each stimulus repeated 10 times in each block with a total of 160 trials, 80 for position testing (20 trials for each position) and 80 for sequence testing (40 trials for each sequence). The trial order was fully randomized with a constraint that the same stimulus would not appear consecutively. No feedback was provided. After the task, as in the first study, participants were required to write down the structure sequences in the right order, and all participants were 100% correct with no error.

3.5.3 MEG Acquisition and Pre-processing

The same procedures for MEG acquisition and preprocessing were applied to both studies. MEG was recorded continuously at 600 samples/second using a whole-head 275-channel axial gradiometer system (CTF Omega, VSM MedTech), while participants sat upright inside the scanner. Participants made responses on a button box using four fingers as they found most comfortable.

The data were resampled from 600 to 100 Hz to conserve processing time and improve signal to noise ratio. All data were then high pass filtered at 0.5 Hz using a first order IIR filter to remove slow drift. After that, the raw MEG data was visually inspected, and excessively noisy segments and sensors were removed before independent component analysis (ICA). An ICA (FastICA, http://research.ics.aalto.fi/ica/fastica) was used to decompose the sensor data for each session into 150 temporally independent components and associated sensor topographies. Artefact components were classified by the combined inspection of the spatial topography, time course, kurtosis of the time course and frequency spectrum for all components, Eye-blink artefacts exhibited high kurtosis (>20), a repeated blink structure in the time course and very structured spatial topographies; Mains interference had extremely low kurtosis and a frequency spectrum dominated by 50 Hz line noise. Artefacts were then rejected by subtracting them out of the data. After that, all later analyses were performed directly on the filtered, cleaned MEG signal, in units of femtotesla.

3.5.4 MEG Analysis

Lasso-regularized logistic regression models were trained separately for the sensory and structure level representations of stimuli. Only the sensors that were not rejected across all scanning sessions in the preprocessing step were used to train the decoding models. A trained model *k* consisted of a single vector with length of good sensors n + 1: slope coefficients for each of the good sensors together with an intercept coefficient.

In both studies, decoding models for the sensory level representation were trained on MEG data elicited by direct presentations of the visual objects. These presentations were taken from the functional localizer task, specifically the MEG data at 200 ms following stimulus onset. This 200 ms time point was selected based on observations from our previous work showing that when object representations are retrieved, the reinstated spatial pattern is most similar to that observed 200 ms after onset of direct object presentation [57] (see also Figure 3.9a). The decoding models for sensory representation were verified on training data through leave-one-out cross-validation approach (Figure 3.9 & Figure 3.5).

The design of Study 2 enabled us to dissociate between the neural representation of sensory and structure level of stimuli. To check whether these classifiers learned abstract information about position and sequence, rather than relying on coincidental sensory features of the stimuli, we used a special cross-validation approach. Instead of holding out individual trials at random, we held out all the trials of one randomly selected object. This meant that a classifier that focused on sensory features would result in below-chance accuracy. To perform above chance, the classifier must identify features that represent the structural information (position or sequence). Using a permutation-based threshold, which corrected for multiple comparisons across time, we found that cross-validated decoding accuracy exceeded chance for both position and sequence code classifiers (Figure 3.9b, c). Accuracy peaked at 300 ms post stimulus onset for the position code, and at 150 ms post stimulus onset for the sequence code.

To explore which MEG sensors contained the most information about object-level and abstract-level representations of stimuli, we repeatedly performed the same cross-validation analysis described above, but each time using a different random subset of 50 sensors. On each iteration, we found a classification accuracy. After performing this analysis 2000 times with different random subsets, we averaged all the accuracies that each sensor participated in, to obtain an approximation of that sensor's contribution to predicting the labels (Figure 3.9).

We then used these trained models to make predictions as to whether unlabelled MEG data corresponded to a neural representation of the state (stimulus/position/sequence) *k*. Each time point was treated independently. At each time point in the unlabelled data, the data vector over sensors was multiplied by beta estimates over sensors of *k*. This procedure yielded a matrix X with number of states of columns and as many rows as time bins.

3.5.5 Sequenceness Measure

We used TDLM described in Section 2 to measure the degree to which decoded human MEG activity tends to follow the transition matrix of the task systematically in either a forward or reverse direction. First the decoding models were trained to recognize each visual object or position, then they were used to decode each stimuli/position reactivation strength during resting/decision-making. For example, state A admits a transition to state B. If the resting period contained forward sequences, then the decoded probability of state A at time T should be correlated with the decoded probability of state B at time T + t, where t defines a lag between neural state representations. This method was validated on simulated data and has been successfully applied.

Evidence of sequenceness was reported as the subtraction between forward and backward direction of same transitions at each time lag. Subtraction in this instance circumvents a potential (auto)correlation shared by both directions at the same time lag and protected the validity of statistical inference. In the current study, the transition matrix was pre-specified based on either the rule-defined order (i.e., “structure sequence”) or experienced visual order (i.e., “visual sequence”).

We also computed the extent to which neural sequences followed longer steps (e.g., length-3, length-4) with the same state-to-state time lag, while controlling for evidence of shorter length. In doing so, we avoid a possibility of false positives arising out of strong evidence for shorter length. The method is largely the same as the GLM approach described above. In addition, we put shorter length transitions into the design matrix as confounding regressors, e.g., if we look for the evidence of A->B->C at 50 ms time lag, we will regress state decoding vector A with time lag 100 ms onto C while controlling for evidence of state decoding vector B reactivated at time lag 50 ms. This process can be generalized to any number of steps. In the current study, we employed a linear track structure, the maximal length is 4, since the end of sequence did not go back to the start of sequence.

3.5.6 MEG Source Reconstruction

All source reconstruction was performed in SPM12 and FieldTrip. Forward models were generated on the basis of a single shell using superposition of basis functions that approximately corresponded to the plane tangential to the MEG sensor array. We calculated the probabilities of replay happened at each time point during rest by multiplying the decoded probability of the first state by the time-shifted probability of the second state. The shifted time is the time lag that was shown to give the peak evidence of sequenceness. After that, we epoched the resting data around the time point that has a high probability (top 5%) to give rise to replay with a time window from -100 ms to 200 ms after the onset of replay.

Linearly constrained minimum variance beamforming [92] used to reconstruct the epoched MEG data to a grid across MNI space, sampled with a grid step of 5 mm. The sensor covariance matrix for beamforming was estimated using data that was bandpass filtered to a broad band, 1-45 Hz, using 0% regularization. The baseline activity is the mean neural activity averaged over -100 ms to -50 ms before the replay onset. All non-artefactual trials were baseline corrected at source level. We looked at the main effect of the initialization of replay. To explore the difference of source between the replay of reward and neutral sequence in first study, we applied the above approach separately for reward and neutral sequence and performed the contrast between reward and neutral trial at source level.

All source images were thresholded at $t > 4$ (uncorrected) for display purposes. Non-parametric permutation tests were performed on the volume of interest to compute the multiple comparison (whole-brain corrected) P-values of clusters above 10 (number of voxels), with the null distribution for this cluster size being computed using permutations (n = 5000).

4 HUMAN REPLAY AND MODEL-BASED LEARNING

4.1 Introduction

Effective learning requires incorporating new experience into our existing knowledge of the world. When you encounter a traffic jam at a crossroads, you learn that the route just taken should be avoided in the future, but equally the value in avoiding the alternate paths that converge to this location. Learning from direct experience can be straightforwardly achieved via "model-free" mechanisms that detect co-occurrence between actions, like routes taken, and subsequent rewards [114,115]. However, it requires additional computation to propagate that experience to more distal situations, as in the example of alternate converging roads, where it may have relevance to future action. Despite behavioral evidence for this type of indirect value learning, we understand little about how it is achieved in the brain [114,115].

In reinforcement learning (RL) theory [6], non-local value propagation can be achieved by "model-based" methods that rely on a learned map or model of the environment to simulate, or simply retrieve, potential trajectories [12]. These covert trajectories can stand in for direct experiences and span the gaps between actions and outcomes [32], a process referred to as experience replay. A potential neural substrate for this process is the phenomenon of hippocampal "replay", where cells in the rodent hippocampus that encode distinct locations in space fire sequentially during rest in a time-compressed manner, recapitulating past or future trajectories [13-15]. Utilizing methods developed to measure neural sequences noninvasively [116] (detailed in Section 2), such replay has now been found in humans during rest [31,54,117], with strong parallels to observations in rodents [54]. Although these events appear appropriate to support value learning, there is little evidence in either species about their involvement.

If experience replay can be shown to support value learning, then its statistics should also bear on a second unresolved question. Given limited time, which of the myriad possible future actions should the brain prioritize to replay? A reward-maximizing agent should prioritize replay of whichever past experiences are most likely to improve future choices and thereby earn more reward. Theoretical analysis [34] argues that such rational priority for replay can be further decomposed into the product of two factors, namely *need* and *gain*. *Need* captures how frequently a given experience will be encountered again in the future, while *gain* quantifies the expected reward increase from better decisions if that experience is replayed.

4.2 Results

4.2.1 Task design

I tested hypotheses that neural replay facilitates non-local learning, and that such replay is prioritized by its utility for future behavior. To detect human replay, I measured whole-brain activity using magnetoencephalography (MEG) while participants performed a novel decision-making task. The task explicitly separates learning from direct vs. non-local experience, permitting the measurement of unambiguous neural and behavioral signatures of the latter.

To isolate local and non-local learning the task comprised three starting states (henceforth called "arms"), each with two alternative choice paths (Figure 4.1A). A choice then leads to a sequence of three stimuli ("path") followed by a final stimulus. At each trial, participants are presented with one of the starting states and asked to make a choice between the two options with the goal of maximizing reward. Importantly, the two end states, reachable from each starting state, are shared across all three starting states. Each end state leads to a reward with a probability that changes slowly from trial to trial. This task structure allows subjects to use reward feedback to inform their choices, including those at other states. The use of three-stimulus sequences allows unambiguous measurement of extended replayed sequences (vs co-occurrence) and their direction.

In addition to distinguishing local (the arm just chosen) from non-local experience, the task allows testing hypotheses that replay and learning favour the higher priority of the two non-local arms. In the design I use, priority differed between arms on the one hand due to *need*, because each starting arm was encountered with a different, but constant, starting probability: rare (17%), occasional (33%), and common (50%) respectively. These probabilities were learnt prior to the main task (Figure 4.1B). Since rewards were stochastic with fluctuating probability, there is also the factor of *gain* from propagating information about outcomes to different arms, and this fluctuated from trial to trial according to individual reward histories. For instance, a reward is more informative in helping to choose actions that would otherwise not be favoured, whereas non-reward is conversely more informative for avoiding actions that would otherwise have been chosen.

I was interested in how participants learn efficiently by incorporating new experiences in the task, particularly those derived from a different starting state (non-local), into updated choices. To achieve this, I first taught subjects a model comprising knowledge of the relations among different elements in the task, as well as the different starting probability assigned to each arm. To avoid biased learning of the model, I introduced each component of the task carefully at different times (Figure 4.1B).

4.2.2 Functional localizer & Model construction

To index neural representations of states in the main RL task, I first showed participants 18 visual stimuli in random order, a task phase called the *functional localizer*. These stimuli were later reused to form distinct states in the RL tasks (e.g., $A1, A2, A3$ in Figure 4.1A). I constructed a probabilistic decoding model for each stimulus based on their evoked neural response in this *functional localizer* task. As before, these decoding models are later used to search for sequential reactivation of states in the main RL task. Notably, these classifiers are unbiased to experiences and task structure, because at the localizer phase subjects had no knowledge of the relationship among those stimuli, nor their value.

The experiment then proceeded across distinct phases to ensure knowledge of the task model (i.e., *model construction,* Figure 4.1B). Thus, upon completion of a functional localizer, subjects learned how the 18 stimuli formed 6 distinct sequences, i.e., the relationship among the 18 stimuli. We refer to this as *sequence learning*. Subjects then learned a mapping between sequences and end states, i.e., *end state learning* and subsequently learned which sequence belongs to which starting arm, i.e., *arm learning*. Up to this point, experience is still unbiased, and participants have learnt the relational structure between arms, outcomes, and sequences alone. At the end of this learning stage, subjects learned the starting probability of each arm, and that these probabilities remained constant throughout the experiment. Subjects also learned the frequency of each starting arm by experience, i.e., *frequency learning*. We ensured that subjects had knowledge of the full task structure with a quiz, where performance was always above 85% (see details in Materials and Methods). Upon completion of the entire training pipeline, participants then performed the main RL task (Figure 4.1A).

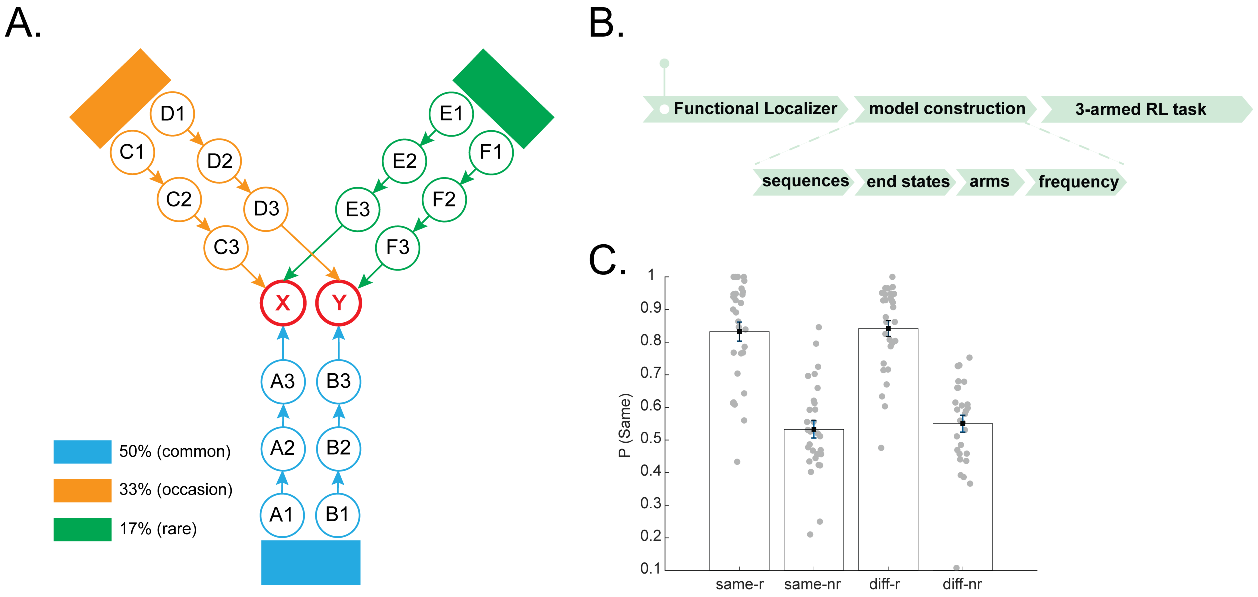


Figure 4.1 Experimental design for model-based reinforcement learning task

(A) The main RL task. At each trial, participants are presented with one of the three starting arms based upon a learnt fixed probability. They then select one from the two alternatives paths within this arm. The reward probability of outcome states changes slowly and independently over time. A crucial task feature is that outcomes (i.e., X and Y) are shared across all three arms, a design feature that enables non-local learning. **(B)** Each experimental task phase is shown. Participants learn a correct task model before commencing the main RL task, and at the beginning, are shown stimuli in a randomized order. **(C)** Behavioral evidence showing exploitation of task model to aid learning. *Same/diff* is defined based on whether the current starting arm is the same or different to that of the last trial; *r/nr* indicates whether subjects were reward or not rewarded at the last trial. P(same) is the probability that participants, in current trial, select a path leading to the same outcome state as that in the last trial. Error bars shows the 95% standard error of the mean, each dot indicating results from each subject.

4.2.3 Behavioral evidence of prioritization in non-local learning

In the RL task, participants need to learn the value of each action at each starting arm, with the aim of maximizing reward. We test whether participants exploit a learnt task model, ascertaining whether they transfer the value obtained in a chosen path to other nonlocal paths that lead to the same outcome. For example, simple model-free learning allows subjects to repeat a previously rewarded action when they encounter the same starting state again. Indeed, we found that, when the next starting state was the same, the participants were more likely to choose the path leading to the same outcome if rewarded on the last trial, compared to no reward (Mixed effects logistic regression, p = 7.5×10^{-15}). However, achieving equivalent learning at the other starting states requires additional model-based computation, such as replay to propagate the reward. In fact, model agnostic data showed the path leading to a rewarded end state was favoured even when the choice was next presented at a different starting state ($p = 9.5\times10^{-23}$), and this effect did not differ between the same and different starting arms ($p = 0.90$ for the main effect of arm, $p = 0.46$ for the interaction effect between arm and reward, Figure 4.1C). This is a hallmark of non-local, model-based learning [118,119].

Next, I used a more detailed computational modelling of the trial-by-trial learning process to test whether this behavioral signature of learning from non-local outcomes was greater for an arm with higher priority. I fit behavior to a computational Q-learning model that updates the value of each arm from obtained rewards (see Methods for modelling details). I augmented this baseline model with additional free parameters measuring the strength of non-local learning as a function of two partial proxies for priority, *gain* (informativeness about choice) and *need* (arm probability) separately. In the task, there are always two nonlocal paths sharing the same end state with the current chosen one, allowing us directly to compare learning across these two non-local paths. We calculated the strength of learning by estimating separate learning rates for the higher and lower priority paths on each trial, with a third learning rate for updating the local (just chosen) arm ($\alpha_d = 0.63$). Numerically, a higher learning rate was estimated for both higher-gain (α_h= 0.79 vs α_l= 0.37, Table 4.1) and higher-need arms (α_h= 0.61 vs α_l= 0.54, Table 4.1), but this difference was significant for gain alone ($p = 0.020$ gain; $p = 0.16$ need, Table 4.1). These results provide behavioral support for a hypothesized rational prioritization of non-local learning. Figure 4.2A, B, see also Figure 4.3

4.2.4 Neural decoding & Sequential reactivations

I next turned to neural data and asked how the observed non-local learning is achieved in the brain. First, I verified that it was possible to decode all 18 visual stimuli well above chance, showing a peak cross-validation decoding accuracy at 47 ± 3 % (vs. chance level, 1/18 ≈ 6%), based on evoked neural response in the functional localizer task (Figure 4.2A, B, see also Figure 4.3, and Methods for decoding analysis details). I then applied the decoding models of these 18 stimuli to the RL task to test for their sequential reactivations at the point of outcome receipt, the time period when new learning occurs. In fact, the focus on this period is analogous to the time when rodents consume a reward (but also see discussion for connections to rodent sequences). We operationally refer to any reactivation of sequences as replay, given we are looking for patterns of spontaneous reactivations off-task.

We first look for spontaneous replay of all possible transitions consistent with the model, where forward sequence expressed the same direction as experience (e.g., $A1 \rightarrow A2 \rightarrow A3$), and backward sequence the opposite (e.g., $A3 \rightarrow A2 \rightarrow A1$). Utilizing the previously described advance in MEG decoding of replay, in this experiment I could now assess the evidence for replay in a forward and backward direction separately [116], found significant forward replay peaking at 30 ms state-to-state time lag (Figure 4.2C), and reverse replay that peaked at 160 ms state-to-state lag (Figure 4.2D, see Methods for sequence analysis details). Consequently, I focus on this 30 ms forward and 160 ms reverse replay in all later analyses.

	Gain model			**Need model**		
Parameters	Mean	5%	$\hat{R}$	Mean	5%	$\hat{R}$
α_d	0.63	0.54	1.00	0.64	0.55	1.00
α_h	0.79	0.65	1.00	0.61	0.53	1.00
α_l	0.37	0.14	1.00	0.54	0.45	1.00
β	1.40	1.20	1.00	1.40	1.10	1.00

Table 4.1 Estimates of free parameters from the gain/need model.

Free parameters: $\boldsymbol{\alpha_d}$ – learning rate for direct experience, $\boldsymbol{\alpha_h}$ – learning rate for non-local experience of high gain (gain model) or need (need model), $\boldsymbol{\alpha_l}$ – learning rate for non-local experience of low gain (gain model) or need (need model), $\boldsymbol{\beta}$ - inverse temperature. mean, 5% confidence interval and the potential scale reduction factor on split chains, $\boldsymbol{\hat{R}}$.

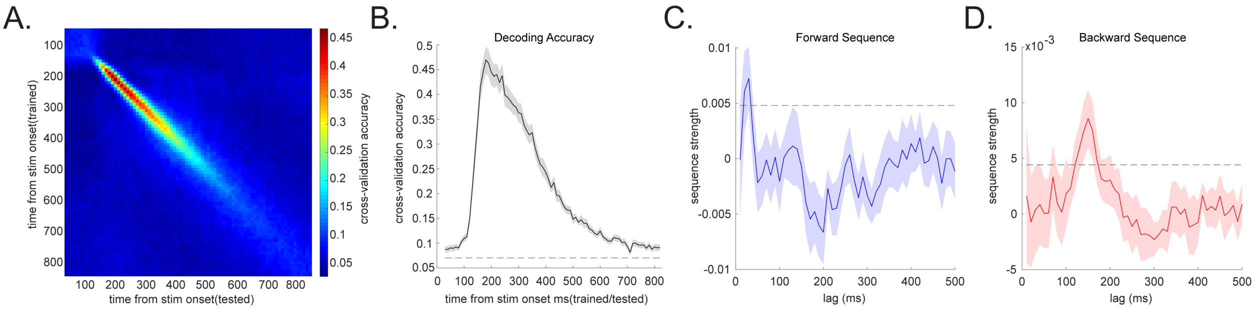


Figure 4.2 Stimuli decoding and neural sequences

(A) All 18 stimuli classifiers are trained based on their evoked multivariate neural patterns at each time bin, from 10 ms to 800 ms post stimulus onset, in a functional localizer phase and tested at all time points across 10-800 ms in a leave-one-out cross validation scheme. This provides temporal generalization plots, with the Y axis indicating time bins the classifiers were trained on, and X axis indicating the test time of classifiers. Accuracy was obtained from all 18 stimuli classifiers, and the readout is deemed accurate if the corresponding classifier of the test label gives the highest decoding probability, as in previous studies [53,54,117]. The diagonal of the temporal generalization plots is the decoding accuracy at the same time we trained the classifiers on, peaking at approximately 200 ms post stimulus onset. **(B)** We trained stimuli classifiers based on their evoked neural response at 200 ms, as in previous studies [54,117]. The dotted line is the permutation threshold taken as the 95% percentile of peak decoding accuracy on randomly permuted labels. (**C-D**) Applying trained classifiers to time of outcome receipt in the RL task. A sequence analysis [116] provided evidence for two distinct sequence signatures, a forward sequence (blue) peaking at a 30 ms state-to-state time lag (C), and a backward sequence (red) peaking at 160 ms time lag (D). The dotted line is the permutation threshold that controls for multiple comparisons. It was taken as the 95% percentile on the peak sequenceness value over all computed time lags in the permutation. This permutation is implemented by randomly permutating the transition matrix, which are shown to be statistically robust [54,116,117]. The X axis is the time lags. Sequence analysis is done separately at each time lag. The Y axis is the evidence of sequenceness, i.e., sequence strength.

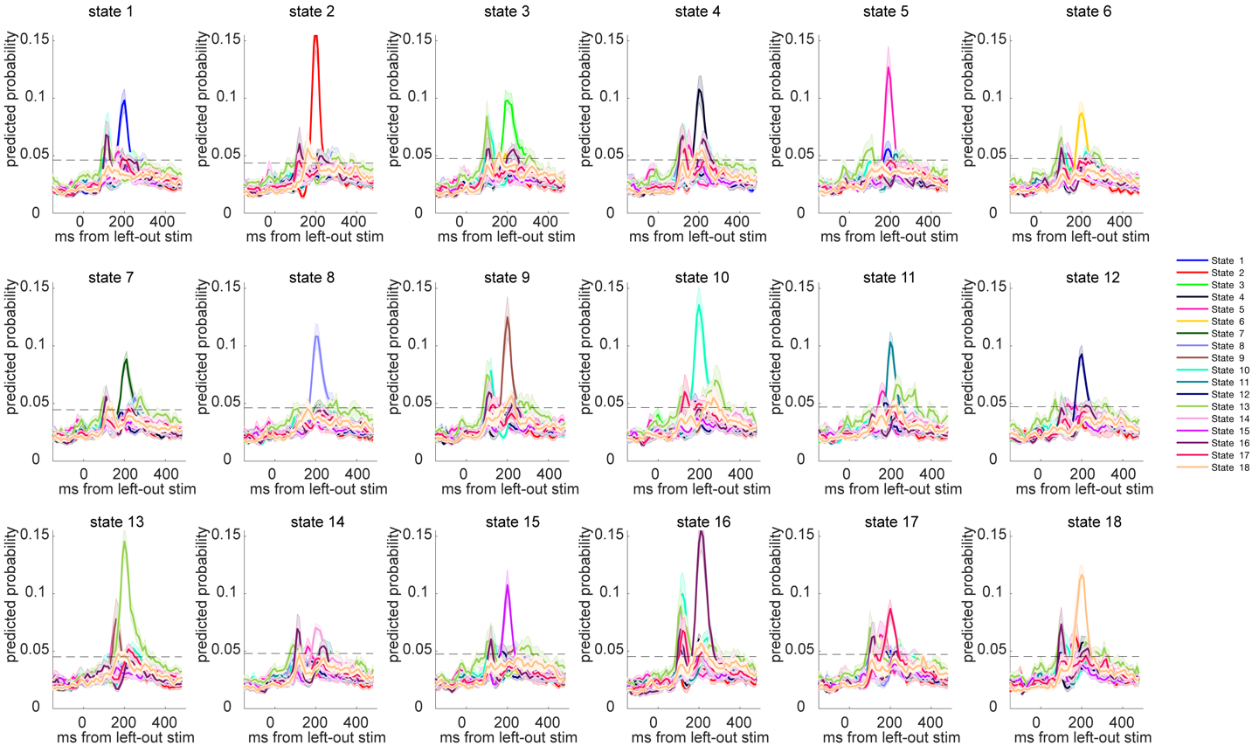


Figure 4.3 Classifiers Performance of the Lasso Logistic Regression Models

The leave-one-out cross-validation results are shown for each classifier (all 18 in total) trained in functional localizer task. Dotted line indicates the permutation threshold estimated by randomly shuffling the labels and redoing the decoding process. These plots only use classifiers trained at 200 ms post stimulus onset, which was used throughout the study for reactivation and sequence analysis.

4.2.5 Two types of replay: functional and physiological differences

The 30 ms state-to-state time lag forward replay is similar to what we found previously during post-task rest [54]. The 160 ms reverse replay has not been reported previously, but is an intriguing finding given its direction is consistent with proposals for solving credit assignment via backpropagating reward, based on theory [34] and empirical data [13,20,54]. Additionally, its slower speed (160 ms state-to-state lag, roughly 6Hz) might be considered to align with a computation capable of supporting online, model-based, cognition. If indeed this 160 ms reverse replay supports non-local updating, then this predicts it will represent contents of non-local trajectories. I found this 160 ms reverse replay dominantly represents non-local paths (one sample *t* test, *t* (28) = 2.92, *p* = 0.007), and does so to a significantly greater degree than local ones (paired *t* test, *t* (28) = 2.21, *p* = 0.03, Figure 4.4A). The 30 ms forward replay shows an opposite pattern (interaction between replay types and representational content, F (1,28) = 15.01, p = 0.001), not significantly representing the non-local paths (one sample t test, t (28) = -0.09, p = 0.93), but mainly the local path, corresponding to recent experience (t (28) = 3.37, p = 0.002, Figure 4.4B).

In addition to a representational difference between the two types of replay, we also tested whether these distinct replay signatures differ in terms of their underlying physiological properties. A previous study [54] showed that fast human replay (with time lag of 20-50 ms) during rest is associated with an increased ripple frequency power (120-180 Hz), akin to sharp wave ripple replay in rodents [22,110,120]. I replicate this finding again, showing that initialization of a 30 ms forward replay is associated with a ripple frequency power increase (one sample t test, t (28) =5.82, p = 2.9×10^{-6}), but this power increase is not present for 160 ms reverse replay (t (28) =0.71, p = 0.48). In addition, there is a significant difference in the power of ripple frequency between the two types of replay (paired t test, t (28) =2.84, p = 0.008, Figure 4.4C). Beamforming results further suggest while both replay types are associated with activation in visual cortex and medial temporal lobe, a 30 ms forward replay has higher hippocampal activation compared to the 160 ms reverse replay, while the 160 ms reverse replay has greater cortical engagement (Figure 4.5).

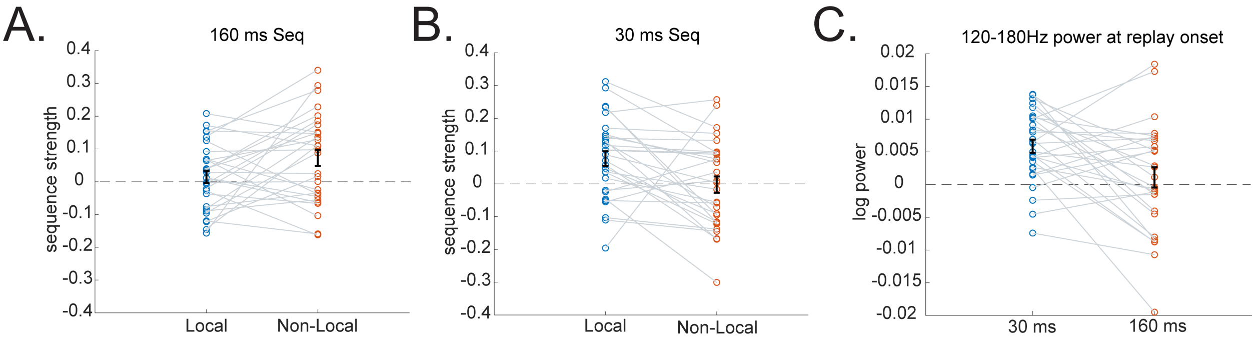


Figure 4.4 Representational and physiological differences between the two types of replay

(A) A 160 ms reverse sequence encodes non-local as opposed to local experience. **(B)** A 30 ms forward sequence encodes local experience alone, but not non-local. **(C)** The initialization of 30 ms forward sequence is associated with a power increase in a ripple frequency band (120-180 Hz), consistent with previous study [54], but this is not the case for 160 ms reverse sequence. These high frequency power signatures are significantly different. The grey line connecting results from the same subject. Error bars shows the 95% standard error of the mean, each dot indicating results from each subject.

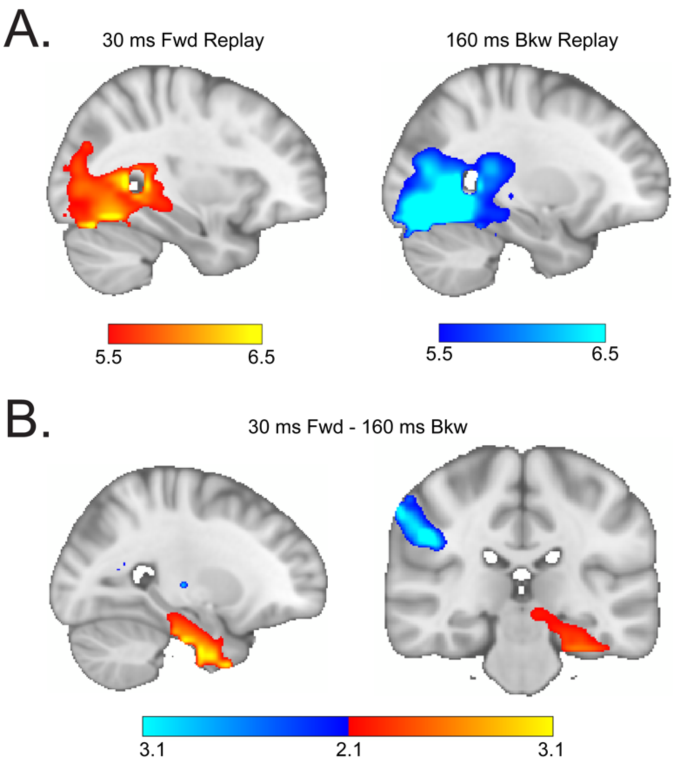


Figure 4.5 Whole-brain source localization of the two types of replay

(A) Source localization of the replay onset for the 30 ms forward replay and 160 ms reverse replay separately, revealed significant visual and medial temporal lobe (MTL, including hippocampus) activation (peak Montreal Neurological Institute [MNI] coordinate: X = -17, Y = –45, Z = –16 for 30 ms forward replay, X = 18, Y = –66, Z = –16 for 160 ms reverse replay). The activation in visual cortex and MTL belong to a similar cluster, and survived whole-brain multiple comparison correction based on a non-parametric permutation test (cluster forming threshold, t = 5, n =5000) for both 30 ms and 160 ms replay. (B) Contrast the 30 ms forward replay vs. the 160 ms reverse replay, which revealed higher activation in the MTL (peak MNI coordinate: X = -26, Y = –10, Z = –29) for the 30 ms replay, and higher cortical regions - postcentral gyrus (peak MNI coordinate: X = 49, Y = –27, Z = 32) for the 160 ms replay. Both the MTL (higher in 30 ms forward replay) and postcentral gyrus (higher in 160 ms reverse replay) survived whole-brain multiple comparison correction based on a non-parametric permutation test (cluster forming threshold, t =2.1, n =5000).

4.2.6 Nonlocal Replay facilitates non-local learning

Having identified replay candidates for learning, I then tested whether non-local replay (i.e., the 160 ms reverse replay) facilitates non-local learning in the service of choice behavior, and if so, whether such replay is competitively prioritized in accord with theoretical accounts [34]. I again posed these questions in terms of RL-based computational models of trial-by-trial choice behavior (see Materials and Methods for modelling details).

First, to ask whether replay helps non-local learning, I augmented the Q-learning model with a term measuring trial-by-trial neural replay. In particular, having first separated learning rates for local and non-local arms (as before), I was in a position to test whether a baseline learning rate for each non-local arm was significantly increased on trials when that arm expressed significant neural replay, vs. when this was not the case. I found higher learning rates in the presence vs. absence of significant 160 ms reverse replay ($\alpha_{replay} = 0.70$; $\alpha_{no-replay} = 0.61$; $p = 0.023$, Table 4.2). This was not the case when, as a control, we repeated the same analysis for the 30 ms forward replay (differce in learning rates = 0.01, *p* = 0.457, Table 4.2), consistent with the lack of representation of non-local arms in the 30 ms forward replay.

Finally, we asked whether replay is prioritized to favour more beneficial experiences. Here I computed a net priority score from the product of *gain* (estimated per-arm, -trial, and -subject from the previous behavioral model) and need (17%, 33%, 50%, for paths in rare, occasion and common arm, respectively) for each non-local path at each trial, comparing replay across the higher vs. lower priority paths on each trial (Figure 4.6A). The priority score is the net product between need and gain. The need is assumed to be the starting probability of each arm, the gain is calculated trial-by-trial based on reward received and related policy change (assuming perfect learning, see details in the Method section). Significantly greater replay was seen for a higher priority path, and this held for reverse replay at 160 ms but not (as expected, for control) for 30 ms forward replay. (*t* (28) = 3.30, *p* = 0.003; for 30 ms forward replay, *t* (28) = -0.34, *p* = 0.74, Figure 4.6B, C). Decomposing this effect, we found no evidence that replay was prioritized according to either *need* or *gain* considered alone (high vs. low need, *t* (28) = 0.17, *p* = 0.87, high vs. low gain, *t* (28) = 0.27, *p* = 0.79).

	160ms reverse replay			**30ms forward replay**		
Parameters	Mean	5%	$\hat{R}$	Mean	5%	$\hat{R}$
α_d	0.65	0.55	1.00	0.65	0.55	1.00
α_{replay}	0.70	0.56	1.00	0.61	0.50	1.00
$\alpha_{no-replay}$	0.61	0.47	1.00	0.60	0.51	1.00

β	1.40	1.20	1.00	1.40	1.10	1.00

Table 4.2 Estimates of free parameters behavioral model

Free parameters: $\boldsymbol{\alpha_d}$ – learning rate for direct experience, $\boldsymbol{\alpha_{replay}}$ – learning rate for non-local experience with replay, $\boldsymbol{\alpha_{no-replay}}$ – learning rate for non-local experience without replay, $\boldsymbol{\beta}$ - inverse temperature. mean, 5% confidence interval and the potential scale reduction factor on split chains, $\widehat{\boldsymbol{R}}$) from the sequence (160 ms vs. 30 ms).

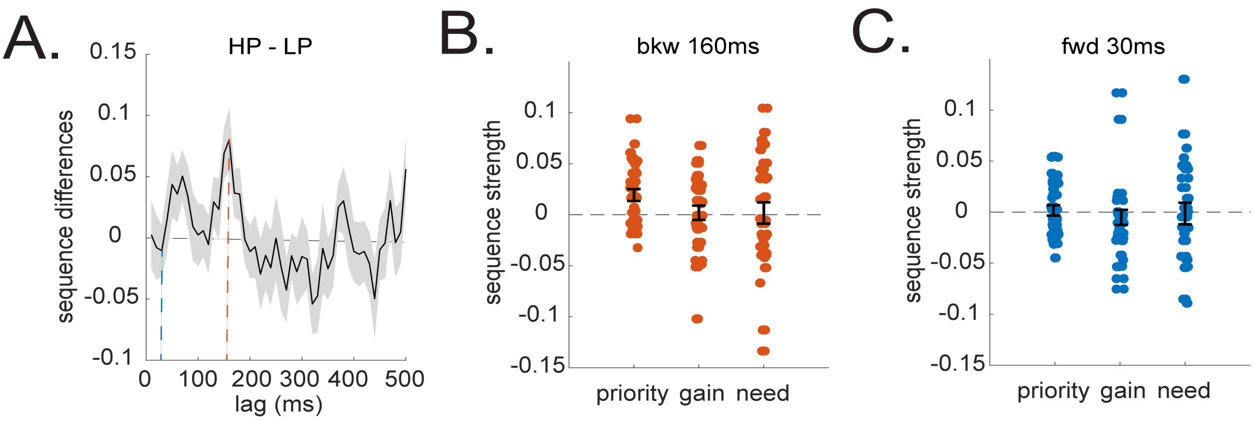


Figure 4.6 Prioritization of non-local replay

(A) Sequenceness differences in high vs. low priority (determined by *gain*×*need*) of non-local paths. We use a backward minus forward sequenceness to provide a summary replay statistic at each time lag. A positive value indicates greater backward sequenceness, while a negative value indicates higher forward sequenceness. The above-zero peak at 160 ms time lag suggests a greater reverse replay for a high priority path. The dotted blue line denotes result at 30 ms lag, and the dotted red line indicates result at 160 ms lag. **(B-C)** Breakdown results plotted separately for *need* and *gain* are shown. The 160ms reverse sequence alone is replayed more in higher priority non-local path. Error bars shows the 95% standard error of the mean, with dots indicating results from each subject.

4.3 Discussion

I show that reverse neural replay, with a 160 ms state to state lag, is a neural signature for non-local value learning. I show this replay facilitates learning of action values by recapitulating non-local experiences that links actions and outcomes across intervening states, evidenced by enhanced learning effects on subsequent choices. I also show that the content of replay, and separately the strength of updating as expressed behaviourally, are prioritized according to their utility for future behavior, consistent with a proposed RL theory [34].

The ability to measure fast sequential replay using MEG in humans, and a disassociation between two types of replay as a function of local vs. non-local learning in current study, establishes for the first time a connection between neural replay and reward learning from non-local credit assignment as expressed in behavior. Accordingly, these findings corroborate a long-standing hypothesis about the role of awake replay on model-based planning and credit assignment [13,20], and extend previous fMRI result in humans linking nonlocal reactivation to planning [118,121].

The 160 ms reverse replay supporting non-local learning is distinct from the forward 30 ms replay reported in previous studies [53,54]. Unlike the latter, the 160 ms replay is not associated with a ripple frequency increase [54], and this raises an intriguing possibility that 160 ms replay, which has a state-to-state frequency of around 6 Hz, might be a homologue of rodent theta sequences [17,45,55]. However, theta sequences generally occur during ongoing behavior in rodents and are forward in direction, akin to a "look ahead" signal. The160 ms replay I identify is (more like sharp wave ripples in rodents [20]) backward in direction and occurs at a trial's end. Whether it corresponds to either the rodent phenomenon is an open question, but the reverse direction and its timing are suited to solve the non-local temporal credit assignment problem, where an outcome at the end of path impacts on decisions made at the beginning.

I conclude that the findings reveal that a non-local replay supports, what is often referred to, as 'model-based learning' via prioritizing memory access. The findings extend the role of replay to account for non-experiential inferential learning and are remarkably consistent with reinforcement learning theory.

4.4 Methods

4.4.1 Participants

31 adults aged 19–31 participated in the experiment, recruited from the UCL Institute of Cognitive Neuroscience subject pool and from a mailing list for MSc students. Eighteen are female, three are left-handed. All participants had normal or corrected-to-normal vision and had no history of psychiatric or neurological disorders. Two subjects were excluded for later analysis. Among them, one was a pilot subject and had gone through a slightly different procedure of task, the other subject had metals in her hair, making her MEG data unusable. This leaves 29 subjects in total for later analyses. All participants provided written informed consent and consent to publish prior to start of the experiment, which was

approved by the Research Ethics Committee at University College London (UK), under ethics number 1825/005.

4.4.2 Details of the task design

The task comprises three phases: functional localizer, model construction and a 3-armed reinforcement learning (RL) task. It is designed specifically to avoid bias in state decoding and ensure correct model of the relational structure has been built before the RL task. The task was implemented in MATLAB (MathWorks) using Cogent (Wellcome Trust Centre for Neuroimaging, University College London).

4.4.2.1 Functional localizer task

All 18 distinct visual stimuli were shown in a randomized order. Those stimuli indicated intermediate states in the main RL task, which form 6 sequences with no overlapping representation (e.g., $A1 \rightarrow A2 \rightarrow A3$; $B1 \rightarrow B2 \rightarrow B3$). Note, the mapping between stimuli and states are randomized among subjects. Participants do not know which stimuli will indicate which state at this time. We will train a classifier for each stimulus based on their evoked neural response in this task and use it to decode state reactivations in later RL task.

At each trial, a visual stimulus (e.g., house) was presented at the centre of the screen for 800 ms, participants were asked to think about its semantic meaning, i.e., this is a house. After that, a text (e.g., "house" or "face") will appear, and the subjects were asked to make a yes or no response within 1000 ms using a response box (mean accuracy: 97.3 ± 0.4%, Mean ± SE). This is followed by a jittered 500-700 ms inter-trial interval (ITI). There are 50 trials for each stimulus presentation, resulting in 900 trials in total. This task is designed to encourage semantic representation of the stimuli, which we have found useful in previous studies [54,117].

4.4.2.2 Model construction task

We want to study model-based learning. We need to make sure the participants have learnt the correct model first. We teach subjects the knowledge of task structure in the following order: 1) *sequence learning*: stimuli-sequence mapping, 2) *outcome learning*: sequence-outcomes mapping; 3) *arm learning*: sequence-arms mapping, 4) *frequency learning*: frequency of the three different arms.

In the *sequence learning*, participants learnt how the 18 stimuli form 6 distinct sequences. Each sequence comprises 3 stimuli. The 3 stimuli appeared on the screen in the right order (e.g., $A1 \rightarrow A2 \rightarrow A3$), with each stimulus lasting for 1000 ms, and followed by the next one. There are 3 learning blocks.

Each sequence was presented 10 times within each block. To test whether participants have learnt the transitions within sequences, we probed their knowledge in the end of each learning block. At probe trial, all three stimuli from the same path were presented, but in a scrambled order. The subjects were required to select those stimuli in the order of true sequence. The mean accuracy is 98.0 ± 0.8%. This was to test their knowledge *within* sequence. We have also tested their knowledge *between* sequences. This time, only one stimulus was presented on the screen, and participants were asked to think about which path does it belong to, for 1000 ms. After that, two alternative stimuli were presented, one from the same path, another was drawn from different paths. Subjects were required to choose the one that belong to the same path. The mean accuracy is 92.1 ± 0.9%. These results suggest the participants have learnt the mapping between stimuli and sequences.

In the *outcome learning*, participants learnt which sequence leads to which outcome. It is crucial that subjects understand the outcome was shared so that the half of all paths leading to the same outcome state, X, while the other half end up in outcome state, Y. During learning, stimuli in each sequence were shown sequentially, and followed by the outcome state. Each sequence and outcome mapping were repeated 10 times. After that, we tested their knowledge. At the probe trial, one sequence (consists of three stimuli) was shown on the screen, and the subject was required to think about which outcome state does it lead to for 1000 ms. Two outcome sates were then shown on the screen, they needed to select the correct outcome state within 1000 ms. The mean accuracy is 97.1 ± 0.8%. This was to test the mapping from sequence to outcome. We have also tested the mapping other way around, i.e., from outcome to sequence. In this probe, the outcome state was shown on the centre of screen for 1000 ms and the subjects were required to think about which sequences can lead to this outcome. After that, two sequences were shown on the screen, these two sequences were associated with different outcomes, the participants were asked to select the one that lead to the same outcome within 1000 ms. To avoid participants only relying on single stimulus in the sequence to make the choice, one out of three stimuli in the sequence was randomly blocked at each prob trial, so that participants were encouraged to think about the sequence as a whole. The mean accuracy is 94.1 ± 0.9% on this probe. These results suggest the participants have learnt the mapping between sequences and outcomes.

In the *arm learning*, participants learnt which sequences belong to which arms. There were three starting arms, each arm has two sequences. The three starting arms also have different encountering frequency in the main RL task, but for now, all arms were experienced equally. The participants did not know which arm they would encounter often or rare, this will be learnt in the next phase.

In this phase, the learning procedure were similar to *outcome learning* phase. After learning, the participants were tested on both the mapping from sequence to starting arms (mean accuracy is 95.4 ± 1.0%), and from starting arms to sequences (mean accuracy is 92.2 ± 1.0%), suggesting they have also learnt the mapping between sequences and starting arms.

In the *frequency learning*, participants learnt the encountering frequency associated with each starting arm. This was fixed and not chosen by the subjects. The mapping between arms and frequency was randomized across subjects but fixed within subjects. It indicated how likely each trial might start in given arm. These different starting probabilities aimed to create different *need* level in the RL task. This is like successor representation in reinforce learning literature [122]. I told the participants explicitly the staring probability of each arm, i.e., rare - 17%, occasion - 33%, common - 50%. I also let participants experience the probability differences of the three arms by showing the three arms according to their encountering frequency in the later RL task. After that, participants were quizzed on the mapping between arms and their starting probabilities. The mean accuracy is 87.9±2%, suggesting they have learnt the frequency differences among the three starting arms.

4.4.2.3 Three-armed RL task

After the subject have learnt all the necessary knowledge about the task structure, they can finally preform the main RL task. There are 5 blocks in this task, each block contains 60 trials, resulting in 300 trials in total. At each trial, a stating arm was shown first, based on its starting probability. The arm picture appeared for 2500 ms, during which, the participants were required to think about which path in this arm they want to choose. After that, the first stimulus of two paths in this arm were shown on the screen (e.g., *A*1 & B1), participant had up to 1000 ms to make the choice. After decision, the chosen path was played out, with each stimulus in that path appearing sequentially on the centre of the screen for 500 ms. The sequence presentation was followed by the outcome state, participants were asked to press the "advance" key to reveal the value (£1 or 0) associated with the outcome. This is to disassociate visual offset of the outcome picture from the credit assignment period. The value display lasted for 2500 ms and followed by a jitter 500-700 ms ITI. The output of the two outcome states was binary and independent from each other. The reward probability for each outcome state follows a Gaussian random walk, with zero mean and 0.2 standard deviations, bounded between 0.25 and 0.75, similar with previous studies [119,123].

At the end of each block, I also tested their knowledge about transitions within sequences to see whether they have forgotten those transitions along the experiment. Participants were shown with three stimuli from the same sequence, but in a scrambled order, and they were required to select the stimuli in the right order of sequence. They were tasked to do so for each sequence twice after each RL block. The mean accuracy is 96.2±0.5%, and it does not change as a function of blocks ($F\,(4{,}112) = 0.89, p = 0.46$), suggesting the sequence knowledge was preserved and remained the same across the whole RL task.

4.4.3 MEG Acquisition and Pre-processing

Whole brain neural activity was recorded using magnetoencephalography (MEG) throughout the experiment, except the time when the participants were experiencing the frequency of different starting arms prior to the RL task. MEG was recorded continuously at 1200 samples/second using a whole-head 275-channel axial gradiometer system (CTF Omega, VSM MedTech), while participants sat upright (3 sensors not recorded due to excessive noise in routine testing). The task was projected onto a screen suspended in front of participants, and participants made responses on three buttons of a MEG-compatible button box, indicating "up/left," "down/right," and "advance" respectively.

Preprocessing was conducted separately for each scanning session, identical to our previous study [54]. Sensor data were high-pass filtered at 0.5 Hz using a first order IIR filter to remove slow-drifts. Data were then resampled to 100 Hz (decoding, reactivation and sequenceness analysis) and 400 Hz (time-frequency analysis), and excessively noisy segments and sensors removed before independent component analysis (ICA). ICA (FastICA, http://research.ics.aalto.fi/ica/fastica) was used to decompose the sensor data for each session into 150 temporally independent components and associated sensor topographies. Artefact components (e.g., eye blink and mains interference) were classified by automated inspection of the spatial topography, time course, kurtosis of the time course and frequency spectrum for all components. Artefacts were rejected by subtracting them out of the data. All analyses were performed on the filtered, cleaned MEG signal at whole-brain sensor level.

4.4.4 Behavior Analysis

To test whether participants have learnt in a model-based way in the RL task I analysed choice behavior as a function of reward in last trial, and same/different starting arm at current trial. The choice at current trial is defined as "same" if the chosen path is leading to the same outcome state (e.g., X) as last trial. We calculate the probability, $P(same)$, as the number of "same outcome" choices divided by the number of all choices minus one. I ask whether it differs based on reward received on last trial (£1 or 0), and the starting arm on current trial (same or different compared to last trial). $P(same)$ under the same starting arm indicate learning from direct experience, while $P(same)$ in the different starting arm measures learning from non-local experience. If the participants have behaved in a pure model-free way, they would not be able to use the outcome in last trial to inform decision in the current trial if it is in a different stating arm, as a result, $P(same)$ under a different starting arm on the current trial would be indifferentiable to reward information at last trial, while the opposite would happen if the participants indeed use the model to learn action and outcome. The results are shown in Figure 1c, consistent with model-based learning account. This logic is the same with similar analysis on two-step decision-making task that set out to test model-based vs. model-free choices [118,119]. We have also simulated the choice behavior following the same reward schedule and starting arms set up in the RL task, by using a model-free vs. model-based Q learning model [118,119]. The simulation results support this reasoning.

The formal statistical analysis is done by fitting a generalized linear mixed-effects model on binary choice behavior (coded as 1, if it leads to the same outcome state, or 0, if that leads to a different outcome, relative to the previous trial), which assume to be binominal distribution. For each trial, the dependent variable – behavioral choice (coded as 1, if it leads to the same outcome state, or 0, if that leads to a different outcome, relative to the previous trial) was explained in terms of reward and starting arm: whether reward was received in the last trial, whether current starting arm was the same with that in the last trial, and the interaction of these two factors. The intercept, and the regression coefficients for reward, arm, and their interaction were all taken as random effects (allowed to vary across participants).

4.4.5 Behavior Modelling

To test whether prioritization happens during model-based learning, I built computational models based on RL theory [34]. At each trial of this task, there is one direct experienced path, and two non-local paths that leading to the same outcome. In the model, I wanted to test whether learning from non-local experiences was elevated in higher need/gain path compared to low need/gain path. I therefore set up three free parameters in the model for the updating rule: learning rate for direct experience, α_d; learning rate for non-local path of high need/gain, α_h; and learning rate for non-local path of low need/gain, α_l. Following Q learning, the updating rules are:

local learning: $$Q(s_d, a_c) \mathrel{+}= \alpha_d(r - Q(s_d, a_c)) \tag{1}$$

nonlocal learning (high need/gain): $$Q(s_h, a_c) \mathrel{+}= \alpha_h(r - Q(s_h, a_c)) \tag{2}$$

nonlocal learning (low need/gain): $$Q(s_l,\ a_c) \mathrel{+}= \alpha_l(r - Q(s_l, a_c)) \tag{3}$$

where the s_d, a_c is the starting arm and action taken at current trial, respectively. Here the action is defined in terms of the outcome state it leads to, so that, if one takes the same action, i.e., a_c in other nonlocal arms (e.g., s_h or s_l), they will end up in the same outcome state. The updating only happens in paths that leading to the same outcome state, because the reward schedule for each outcome is independent, getting reward (or not) at X, does not tell us anything about Y.

The s_h is a nonlocal arm where it contains the higher need or gain path, and s_l is the other non-local arm. In the prioritization based on *need*, this calculation is straightforward because the starting probability for each arm is fixed and known to the participants, therefore s_h is the arm that has higher starting probability than s_l. In prioritization based on *gain*, this has to be calculated trial-by-trial because the reward probability for each outcome state is changing gradually over trials. The gain is defined as the policy gain in replaying a given piece of experience. This can be written based on Mattar and Daw [34]:

$$Gain(s_k, a_c) = \sum_{a \in A} Q_{\pi_{new}}(s_k, a)\pi_{new}(a|s_k) - \sum_{a \in A} Q_{\pi_{new}}(s_k, a)\pi_{old}(a|s_k) \tag{4}$$

Where s_k indicate the non-local arms, $\pi_{new}(a|s_k)$ represents the probability of selecting action a in state s_k after the Bellman backup, and $\pi_{old}(a|s_k)$ is the same quantity before the Bellman backup. In this task, there are only two actions available in A: a_c and $\neg a_c$, where $\neg a_c$ indicate the action that leads to different outcome state.

Assuming perfect learning and a SoftMax policy rule, the above Equation (4) can be re-written based on reward information in the current trial:

Reward: $$Gain(s_k, a_c) = 1 - [\pi_{old}(a_c|s_k) + Q_{\pi_{new}}(s_k, \neg a_c)\pi_{old}(\neg a_c|s_k)] \tag{5}$$

No Reward: $$Gain(s_k, a_c) = Q_{\pi_{new}}(s_k, \neg a_c) - [0 + Q_{\pi_{new}}(s_k, \neg a_c)\pi_{old}(\neg a_c|s_k)] \tag{6}$$

Where $Q_{\pi_{new}}(s_k, \neg a_c)$ is the same as $Q_{\pi_{old}}(s_k, \neg a_c)$ given no updating happen at the action leading to a different outcome. We can therefore use compare $Gain$ in two non-local arms and assign α_h to higher gain path, and α_l to lower gain path. In the end, we are interested whether $\alpha_h > \alpha_l$.

The decision rule follows a SoftMax policy, with another free parameter, β, that quantify how much the choice is determined based on Q value. Subjects were assumed to generate actions stochastically, according to a sigmoidal probability distribution:

$$P(s_k, a_c) = \frac{1}{1 + \exp(-\beta\,[Q(s_k, a_c) - Q(s_k, \neg a_c)]\,)} \qquad (7)$$

To estimate the model, we utilized Markov Chain Monte Carlo (MCMC) methods, implemented in the Stan modelling language (Stan Development Team). We produced 4 chains of 10,000 samples each. The first 2500 samples from each chain were discarded to allow for equilibration. We verified the convergence of the chains by visual inspection, and additionally by computing for each parameter the 'potential scale reduction factor', $\hat{R}$ [124]. For all parameters, we verified that $\hat{R} = 1.0$, consistent with convergence range [125]. As a sanity check, we have also run parameter recover to ensure the modelling results are not biased due to certain reward schedule, etc.

The model was specified hierarchically, so that the participant-specific parameter estimates were assumed to be drawn from a population-level distribution. The prior for all group parameters are assumed unit normal, with mean $\mu = 0$ and standard deviation $\sigma = 1$. For learning rate, the parameter fitting was done in the untransformed space $(-\infty, +\infty)$ and transformed to [0, 1] for model-based calculation.

4.4.6 Multivariate MEG Analysis: Stimuli Decoding and Sequences

Sequenceness analysis relies on the ability to quantify transient spontaneous neural reactivations of task stimuli. For each stimulus ($k \in [1{:}18]$) indicating intermediate states in the RL task (e.g., $A1, A2, A3, B1, B2, B3$, etc.), I trained a separate lasso-regularized logistic regression model based on their evoked neural response from Functional localizer task, at each 10 ms time bin from 0 ms to 800 ms post-stimulus onset. Each model k discriminated between sensor patterns pertaining to stimulus k compared to all other stimuli plus an equivalent amount of 'null' data from the inter-trial interval. Inclusion of null data reduces the spatial correlation between classifiers, and help sequence detection [54,116]. To quantify classifier accuracy, the models were trained in leave-one-out cross-validation and prediction accuracy estimated as the average proportion of test trials where the classifier reporting the highest probability corresponded to the trial label (Figure 4.2.A and Figure 4.3). The cross-validation accuracy peaked around 200 ms, which similar to our previous studies [54,117], and was also the time bin reported to give the strongest semantic representation [57]. I confirmed that decoding accuracy was significantly greater than chance level using nonparametric permutation test. Specifically, I permuted the labels of test trials 500 times per participant, and for each permutation identified the maximal mean accuracy over participants from 0 to 800 ms post-stimulus onset (controlling for multiple tests over time). Accuracy in the unpermuted data was deemed significant it exceeded 95% of within-permutation maximal accuracies (i.e., dotted line in Figure 4.2 A and Figure 4.3).

I trained the stimuli classifiers based on the whole brain multivariate sensor patten at 200 ms post-stimulus onset (Figure 4.3). These classifiers were used to decode state reactivations in later RL task for sequence analysis. L1 regularization, $\boldsymbol{\lambda}$, was used to encourage sparsity and enhance sensitivity for sequence detection [54,116]. To ensure the results were not overfit to the regularization parameter, we fixed $\boldsymbol{\lambda} = 0.005$ for all subjects, based on sequence results from a pilot subject (data which was not included in formal analysis). On the pilot data, this $\boldsymbol{\lambda}$ value maximize an average sequenceness value across 10 ms to 200 ms state-to-state time lags.

I applied the trained stimuli classifiers to the end of each trial in the RL task, after outcome value receipt, this gave us a time*state decoding matrix. I then used Temporally Delayed Linear Modelling (TDLM) to quantify evidence for sequential reactivations in this decoding matrix [116]. This is the same analysis approach applied in my previous studies, and quantifies sequenceness in a forward and backward direction separately, while controlling for auto-correlation [54,117]. I quantified sequenceness for all possible transitions permitted by the task. Evidence of replay in all 6 paths were estimated at the same time, thereby controlled for common variance. I calculated sequenceness from time lag 10 ms to 500 ms.

Statistical significance was assessed using state-identity based permutation test [116]. The null hypothesis is that the state identities are exchangeable. In these permutations, I measure sequenceness of random transitions that are not consistent with the task structure, e.g., $A1 \rightarrow B3$, also from 10 ms time lag to 500 ms. The permutations were run 1000 times. We determined the significant threshold by first taking the maximum sequenceness in the permutations across all computed time lags (to control for multiple comparisons), and then the 95% percentile on that peak across samples. Any true sequenceness that exceed this threshold is deemed significant. This approach has been validated in previous work, including my own, in both simulation and empirical data [54,116,117].

4.4.7 Sequence - Behavior Modelling

To test whether non-local replays facilitate model-based learning. We again built a computational model to ask does inclusion of replay of given non-local path accelerate learning on that path?

First, we identified significant replay events in each path and at each trial. The significance is determined in the same way as before, except we assess significance separately for each path and each trial, rather than relying on a grand average. We do that separately for 30 ms forward sequence and 160ms reverse sequence. A significant replay event is coded as 1, and non-significant event is coded as 0. We call this variable, seq.

Following the Q learning model, the updating rule is:

local learning: $$Q(s_d, a_c) \mathrel{+}= \alpha_d(r - Q(s_d, a_c)) \quad (8)$$

nonlocal learning: $$Q(s_k, a_c) \mathrel{+}= [\alpha_n + \alpha_r seq(s_k, a_c)](r - Q(s_k, a_c)) \quad (9)$$

Where the s_d, a_c is the starting arm and action taken at current trial, respectively, s_k indicate the non-local arms, same with the behavioral modelling described above. α_d is the learning rate for direct experience, α_n is the baseline learning rate for nonlocal experience, and α_r is the replay modulated learning rate. Other than the updating rule, the decision rule, model prior and fitting procedure were exactly the same with the behavioral modelling described above. In this modelling analysis, we are interested in whether $\alpha_r > 0$.

4.5 Supplementary Analysis

4.5.1 Sequenceness as a function of reward and starting arms

In addition to model-based analysis, I also examined how replay differs when responding to receipt of reward, and whether it is modulated by starting arm probability. I reasoned that if replay is indeed sensitive to gain of informing policy, rather than just prediction error, replay should be stronger in trials when subjects get no reward vs. reward, because with everything else being equal, no reward is more informative. No reward suggests a change of action in the next trial might be desirable, while getting a reward means the subject is already doing the right thing. This reasoning is also supported by modelling results, the probability of higher gain paths in non-local experience is higher for trials when a subject does not get reward ($P(high\ gain|No\ reward) = 0.73$), compared to when getting a reward ($P(high\ gain|reward) = 0.32$). I found the 160 ms reverse replay of non-local paths are indeed higher for no-reward trials compared to reward trials (Figure 4.7). This reward modulated replay is also stronger in high need (common) arm, compared to low need (rare) arm (Figure 4.7). Those results were specific to replay of nonlocal paths leading to the same outcome, with no significant difference evident for paths leading to a different outcome (Figure 4.7). Although alternative explanations might exist, one possible account is that this reward modulated replay is also prioritized based on need.

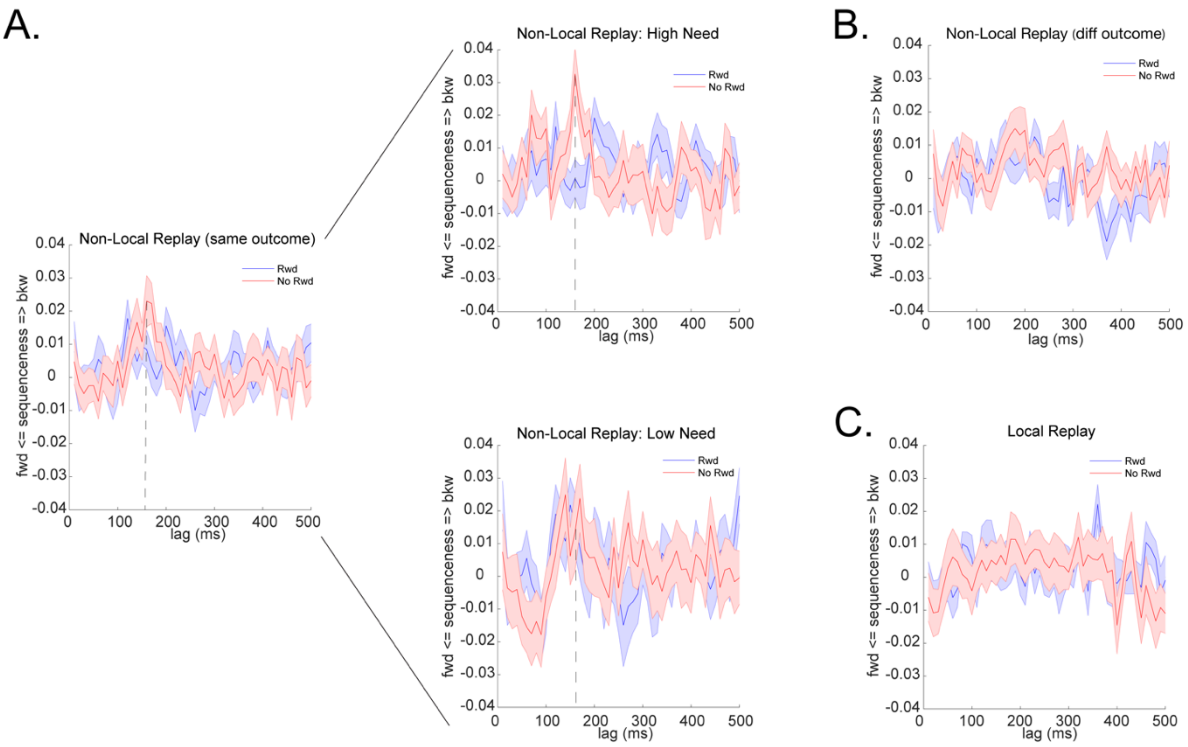


Figure 4.7 Non-local Replay is modulated by reward and need during credit assignment

(**A**) Stronger reverse replay at 160 ms time lag in non-local paths that leading to the same outcome, if this outcome provided no reward vs. reward. This reward modulated replay is greater for nonlocal path that belong to a high vs. low need arm. The X axis is time lag, and the Y axis is the difference between backward (bkw) sequence and forward (fwd) sequence, where positive value indicates higher bkw than fwd replay, and vice versa. The dotted line indicates results at 160 ms time lag, the lag focus in the paper, and also provides the peak evidence for reward modulation. **(B)** This modulated replay is not evident for non-local paths that leading to different outcomes. **(C)** Local replay is also not modulated by reward.

4.5.2 Reactivation analysis

In addition to a sequence analysis, I also examined reactivation alone. In theory, subjects can reactivate the first stimuli of a path during credit assignment time for updating the value, because the current RL task does not entirely depend on sequencing. Recall choice is made on the first state of a path under each arm. This reactivation account has been suggested in previous studies, using a sensory preconditioning paradigm [57,77]. We did not find this effect in our data. Reactivation of the first stimuli of a path did not facilitate learning, nor was it modulated by either reward, choice, or arm (Figure 4.8). I also looked at reactivation of arm pictures during credit assignment time and again did not find reactivation of an arm facilitated learning, nor was it related to reward or choice (Figure 4.8), albeit I can decode all three arms pretty well (with a peak cross-validation accuracy around 54±1.3%, Figure 4.8). I used the same training time (200 ms) and L1 regularization as we have used throughout out the paper (results shown in Figure 4.8). I also tried both L1 and L2 regularization and a wide range of regularization values for the reactivation analysis, but none of these resulted in significant results. It is possible that different types of representation were reactivated. Note here I have not tried training classifiers at different time bins, or with combinations of regularization value [57]. But the null results here indicate my sequential replay results cannot be explained by reactivation alone. It is also possible that although the RL task itself does not require sequencing, the prior extensive learning of task structure and a requirement to always remember relational structure (i.e., sequences) throughout the experiment, constrained subjects to choose a sequential mechanism, rather than a simpler reactivation alone.

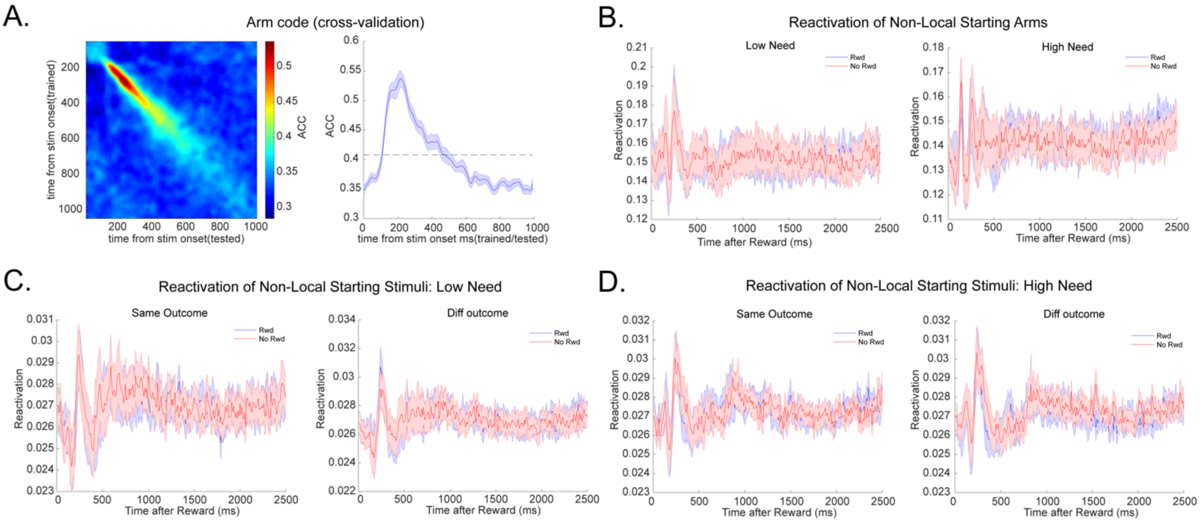


Figure 4.8 Reactivation of the non-local arms and starting stimuli during credit assignment

(A) Temporal generalization and decoding accuracy (leave one-out cross validation) for three arms. We trained the arms classifiers based on the evoked neural response in the quiz of *arm learning* task where only one arm picture was presented on the centre of screen and the participants were required to think about which two paths belong to this arm. We see a similar neural dynamic of the arm representation. We trained the arm classifiers in exactly the same way as the 18 stimuli classifiers. The dotted line the permutation threshold. **(B)** Apply the trained arm classifiers to the outcome receipt time in the RL task, we see the reactivation alone of the non-local arms is not modulated by reward or need. (**C, D**) We have also looked the reactivation alone of the two starting stimuli which the participants choosing from. They are also not modulated by reward, leading to same/different outcome, or need.

4.5.3 Reactivation and sequence analyses in the decision time

In theory, this task can be solved by planning at decision time using either reactivation or sequential replay. The latter mechanism was suggested in a previous fMRI study [118]. At decision time, one can prospectively reactivate the desired outcome state, or sequentially replay the path leading to the outcome. If this is the case, reactivation or replay content should predict what subjects are going to choose in that same trial. I did not find evidence to support these notions. Although I can decode outcome states pretty well (with a peak cross-validation accuracy around 65±1.1%, Figure 4.9), I did not find it predict behavior, nor was it modulated by reward (Figure 4.9). I found no evidence of sequential replay in general at decision time (Figure 4.9), something that might be expected based on Mattar and Daw [34]. There is nothing special in terms of the aims of replay during credit assignment and during planning time and the aim is always to learn an optimal behavioral policy in the most efficient way. Given the whole task here can be solved at credit assignment time alone, i.e., figure out the best action under each starting arm, there is little cognitive imperative as decision time to invoke replay. A difference between Doll, et al. [118] and current results might be due to the fact the current study has distinct representational contents for each path, and it is a one-step decision task, which makes the planning easier to achieve, thereby making this process harder to capture in the data.

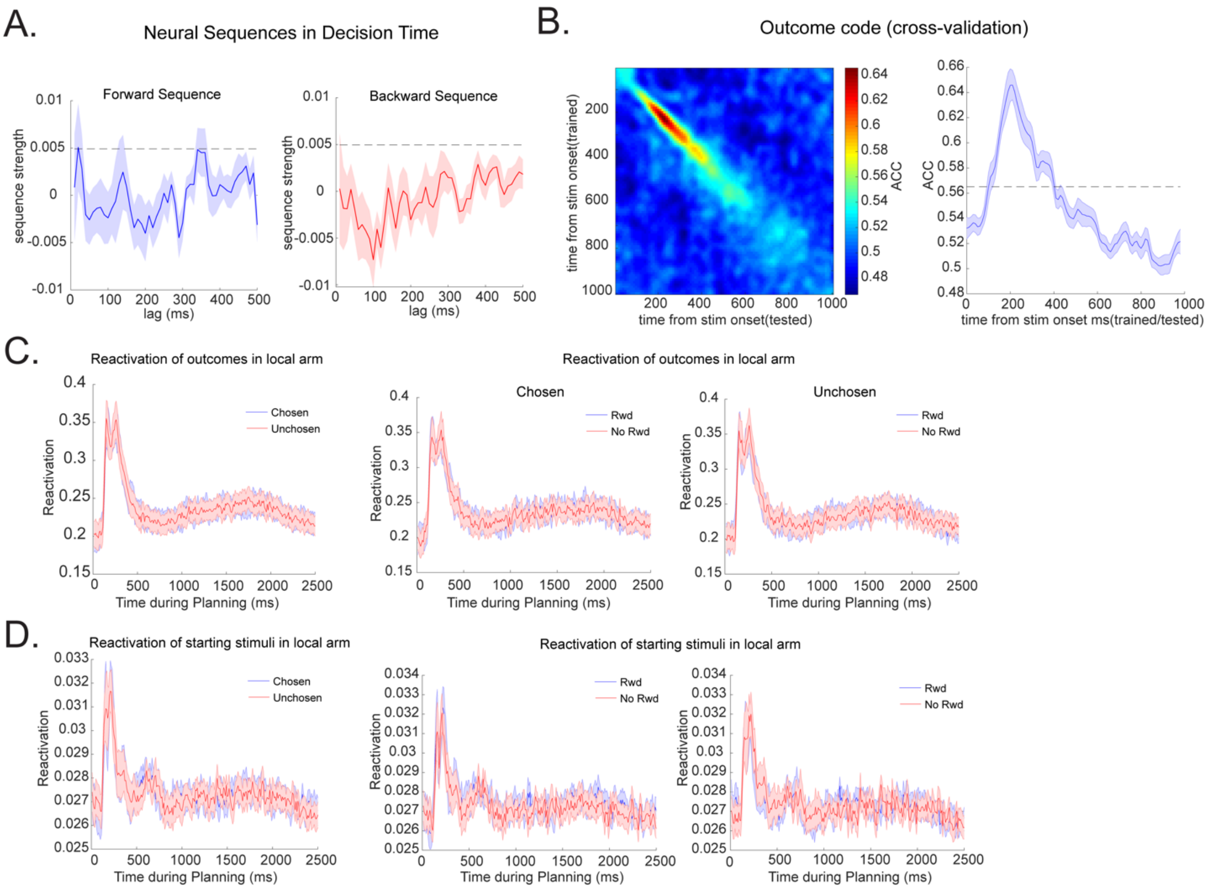


Figure 4.9 Replay and reactivation results in the decision time

(A) No reliable neural sequence effect is seen at decision time. The only sequenceness that just pass the permutation threshold is a 20 ms lag forward replay. It is consistent with a faster replay we found during the credit time, and within a previous study [54]. But it is not modulated by either reward, need or choice. **(B)** Decoding results (leave one-out cross validation) for the two outcomes were shown. We trained the outcome classifiers based on the evoked neural response at the quiz during the outcome leaning task, where the one outcome picture presented in the centre of screen at a time, and subjects were asked think about which paths lead to this outcome. The dotted line is the permutation threshold. We again trained the outcome classifier in the same procedure as the arm and stimuli classifiers. **(C)** Apply the outcome classifiers to the decision time, we show the outcome reactivation is not modulated by reward or choice. **(D)** Likewise, reactivation alone of the two staring stimuli in the current arm was not modulated by choice, reward, as well as whether the arm is high or low need.

5 HUMAN REPLAY SUPPORTS MODEL-BASED PLANNING

5.1 Introduction

Humans are remarkable at adapting their behavior to ever changing situations. We plan to take a detour if we know in advance our usual route is blocked. We mentally navigate complicated problem spaces and decide the next best move when playing a board game like chess or go. Yet we know little about how this can be achieved in the brain [89].

In rodents, theta sequence is hypothesized as important for route planning in space [42,126]. Theta sequence is defined as sequential firing of place cells that are nested within theta rhythm in the local field potential of hippocampus [45] and plays a key role in planning or look-ahead during active spatial navigation. It runs in a forward direction; sometimes predicting the path an animal will run in the immediate future [126]. This is in contrast to typical replay found during rest in rodents, which is associated with sharp-wave ripple (SWR), and can be either forward or backwards [13,110]. Although replay is thought to be part of a mechanism for memory consolidation or learning during rest, evidence suggests it can also play a role during active spatial navigation [17]. In either case, the fast-sequential reactivations are thought to be important. In this final study I test this, again using magnetoencephalography (MEG).

5.2 Results

5.2.1 Task design

MEG is used to detect fast spontaneous neural sequence in humans when subjects are planning sequentially (4-steps) in a non-spatial state space, where states are defined by decodable visual objects. The state space consists of six states where each state transits to, and can be transited from, two different states (Figure 5.1A). The task is designed specifically to encourage sequential planning on a trial-by-trial basis as crucial new information needs to be taken into consideration at each trial (Figure 5.1B, C). Firstly, at the beginning of each trial, two (randomly selected) states are assigned to be ‘‘neg’’, meaning that reaching either of these states multiplies the trial’s cumulative reward by -1 (Figure 5.1B); secondly, the reward amount of each state changed by -1p, 0p, or 1p randomly at each trial (Figure 5.1C). Detailed task description and behavioral analysis can be found in Kurth-Nelson, et al. [53] This project is a re-analysis of the data reported from Kurth-Nelson, et al. [53] but now using the newly developed method -

TDLM described in Section 2. Specifically, in this instance I am looking for sequenceness at the planning time. Before scanning, all participants learnt the task transition structure. They have reached 100% accuracy on a set of automated quiz questions that probed knowledge of the transition structure (e.g., “if you start at cat, and press up, where will you be?”). In post-scanning debriefing, all participants reported a subjective experience of deploying knowledge of transitions for planning, which is also supported by modelling results on behavioral choice data during MEG scan (see details in Kurth-Nelson, et al. [53]).

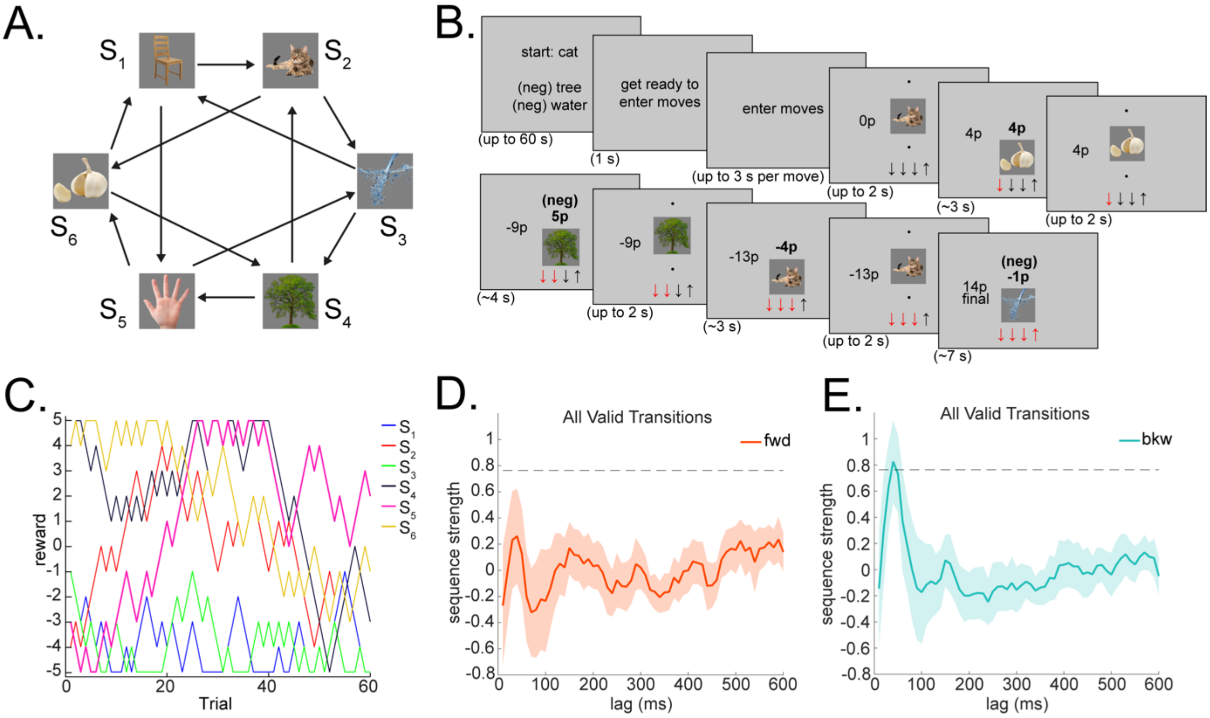


Figure 5.1 Planning task design and forward and backward sequence strength of all valid transition

(**A**) Participants navigated between six states (S1–S6), each corresponding to a visual object. The transitions between states are fixed, but the visual objects assigned to each state number were randomized across participants. (**B**) On each trial, participants began in a random state and were permitted four moves. They had up to 60 s to plan these four moves. The four moves were then entered rapidly with no feedback. After rapidly entering their chosen sequence of moves, participants were required to play out this sequence. While playing out the sequence, the objects and their associated reward were visible. (**C**) The reward associated with each state drifted slowly over trials. The total reward earned in each trial was the cumulative reward collected along the path. When a ‘‘neg’’ state was reached, it caused the sign of the cumulative collected reward to flip (negative to positive and vice versa). (**D**)There was no significant evidence for sequencing in forward direction when assessing sequence strength at each time lag independently from 10 ms to 600 ms (x axis). The sequence strength is defined as the unique predictability of given state to its successive state in the task. The dashed line shows the 95th percentile of state shuffles over all lags, which corrects for multiple lags. Red line indicates the mean forward sequence across subjects. Shading indicates stand error of the mean. (**E**) There is significant sequence strength in a backward direction, peaking at 40 ms time lag. This is a similar result to that obtained using a cross-correlation approach from Kurth-Nelson, et al. [53]. Comparison Results using cross-correlation was shown in Figure 5.3. Blue line indicates the mean forward sequence across subjects. Shading indicates stand error of the mean.

5.2.2 Sequence decoding: Policy vs. Non-Policy

To test for neural sequences that following the transition structure of the planning task, we trained a multi-class classifier on the MEG data recorded 200 ms post-stimulus onset in a different task - functional localizer task. Similar decoding accuracy was obtained as our previous report (Figure 5.2, cf. Figure 2 from Kurth-Nelson, et al. [53]). I then applied the trained decoding models to MEG data collected at the planning phase of each trial. This transformed the MEG data to a state reactivation probability time series. I then choose to apply temporal delayed linear modelling (i.e., TDLM) approach on the state reactivation matrix, which has the advantage of assessing forward and backward sequence separately compared to our previous cross-correlation approach (cf. Figure 5.3 for comparison results). This sequence analysis controls for non-specific dynamics and asks whether on average and to what extent does state i at time lag Δt uniquely predict state j, compared to the evidence for all other transitions. No significant forward sequence of permitted transitions (valid) was found (Figure 5.1D), while a significant backward sequenceness of all permitted transition was observed (Figure 5.1E), peaking at 50 ms state-to-state time lag (between-subject sign flip permutation test, p = 0.030), consistent with a previous report using cross-correlation (cf. Figure 3 from Kurth-Nelson, et al. [53]).

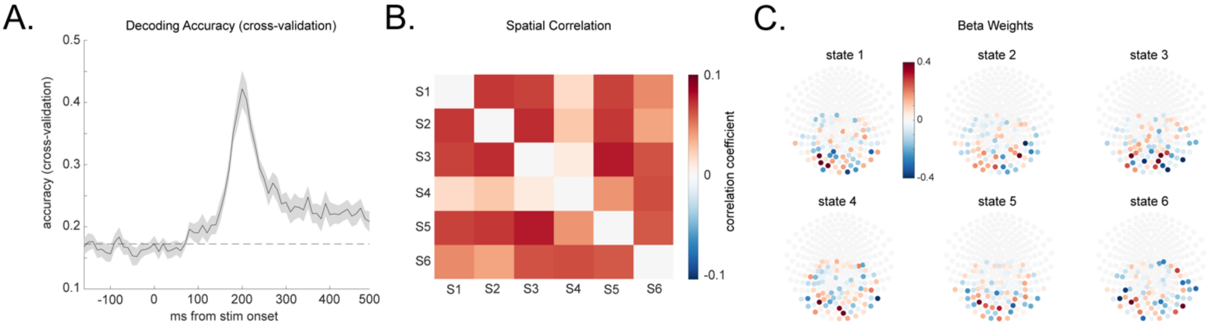


Figure 5.2 Multivariate decoding model

(**A**) Prediction accuracy is estimated by treating the index of the model with highest probability output as the predicted object in leave-one-out cross-validation scheme. The classifier is trained on 200 ms post-stimulus onset and tested from -150 ms to 500 ms in a leave-one-out cross-validation scheme. The prediction accuracy reached 42.2% ± 2.3%, where dashed lines show 95% of empirical null distribution obtained by shuffling state labels. Shading indicates SEM. L1 = 0.6 is selected given it gives the highest decoding accuracy and it is used throughout the analysis for all subjects. (**B**) Spatial correlation between beta weights among all six states, except for correlation with itself (which is always 1). (**C**) Beta weights distribution over sensors separately for all six states.

To look for neural sequences related to planning, at each trial, we estimated sequenceness of transitions that are part of the behavioral planning trajectory (only those steps that were ultimately chosen) – "*policy*" sequence, versus sequenceness of transitions that are valid but not part of the planning path – "*non-policy*" sequence, looked at separately in a forward and backward direction (Figure 5.4). We see only a policy sequence that was played in a forward direction, peaking at 150 ms state-to-state time lag (Figure 5.4A, p = 0.031, see Figure 5.5 for sequenceness breakdown of each pair-wise policy sequence), in addition to backward sequencing of both policy, and non-policy sequence peaking at 50 ms (Figure 5.4C, D, policy sequence, p = 0.026; non-policy sequence, p = 0.041). The forward policy sequence is intriguing, given both in its speed (150 ms ≈ 7 Hz) and direction as these are consistent with theta sequence reported in rodent during active spatial navigation. Further separating the behavioral policy sequence based on whether the transitions belong to optimal policy path or not, revealed this 150ms peak is stronger for a sequence of optimal path (p = 0.004), and is specific to those behavioral policies as no sequence effect was found for transitions that are part of an optimal policy but were unchosen, suggesting the forward 150 ms sequence is a signature of actionable plan.

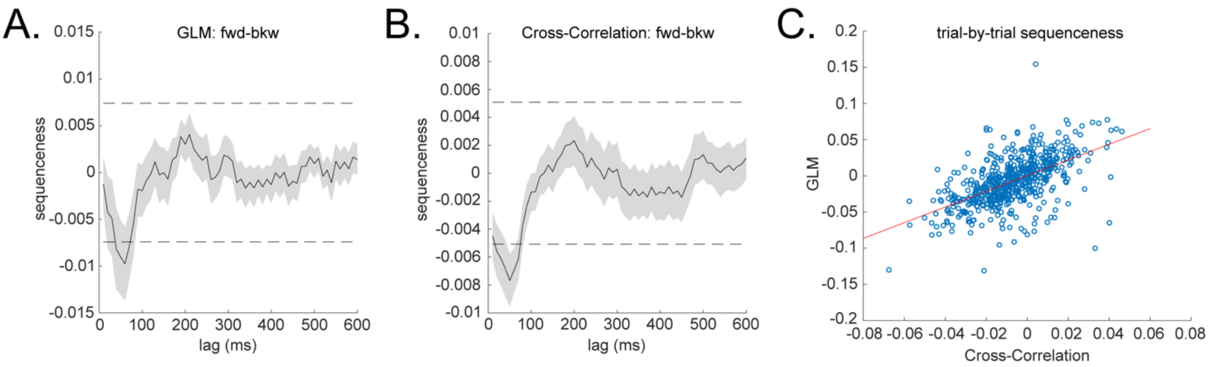


Figure 5.3 GLM vs. Cross-correlation results

(**A**) Sequenceness calculated using the current GLM approach. Sequenceness is defined as the difference between forward and backward sequenceness for the same transitions, as with my previous work. It is assessed independently at each time lag, negative value indicates more backward than forward sequence, and vice versa. The dashed line is permutation threshold, given as the two-tail 95th percentile of the absolute value of shuffles over all lags (10-600 ms). Shading indicates stand error of the mean. The sequenceness is significant from 40-70 ms, peaked at 50 ms. (B) Sequenceness calculated using the original cross-correlation approach. The sequenceness is significant from 20-70 ms, peaked at 50 ms. (C) Scatter plot of sequenceness at 50 ms lag calculated from GLM approach vs. cross-correlation approach. Each dot indicates a trial, it is pooled over all trials across subjects. The red line is the best least-square linear fit.

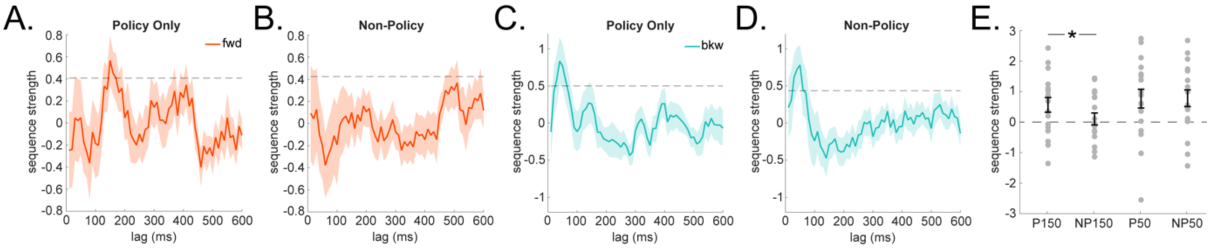


Figure 5.4 Sequence strength of policy-related vs. non-policy but still permitted transitions

(**A**) There is significant forward sequence strength of policy-related transitions peaking at 150 ms lag. The permutation threshold (dashed line) here controls for multiple comparison over 100 to 600 ms time lag, given we are interested in sequence information with state-to-state time lag larger than the 20 -70 ms previously reported. Red line indicates the mean forward sequence strength across subjects. Shading indicates stand error of the mean. (**B**) No significant forward sequence strength of permitted transitions was detected that are not part of a policy trajectory (i.e., non-policy). (**C**) There is significant backward sequence strength of policy-related transitions that peaked at 50 ms lag. Blue line indicates the mean forward sequenceness across subjects. Shading indicates stand error of the mean. (**D**) Similarly, significant backward sequence strength of non-policy transitions peaked at 50 ms lag. Note, while the dashed line indicates permutation threshold over 100 to 600 ms time lag, the backward sequence strength (policy or non-policy) at 50 ms lag can also pass the permutation threshold defined as 95th percentile over all time lags (10-600 ms). (**E**) Forward sequence strength of policy transitions at 150 ms lag is significantly stronger compared to that of non-policy transitions, while the backward sequence strength of policy transitions at 50 ms is not different compared to that of non-policy transitions. Here each dot indicates each scanning session. Error bar indicates stand error of the mean across scanning sessions. This is for visualization purpose. The statistical inference is done by linear mixed model. * indicates statistically significant, $p < 0.05$.

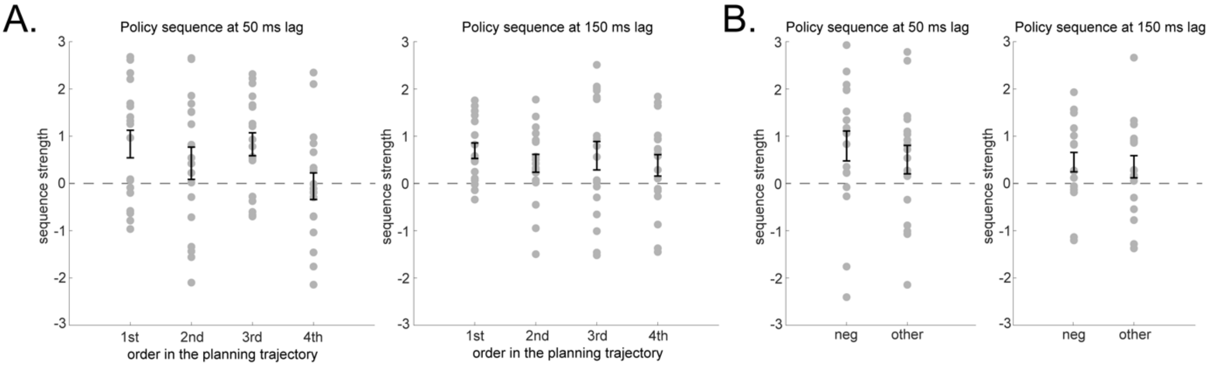


Figure 5.5 Sequence strength of each transitions within the planning trajectory.

(**A**) At each trial, participants need to plan 4 moves ahead. Sequenceness of each move during planning time is plotted stepwise for each scanning session, separately for 50 ms (left panel) and 150 ms time lag (right panel). For instance, the 1st step is the transition from the starting state to second state of the planning path; the 4th step is the transition from the 4th state to the end state of the whole planning trajectory. Each dot indicates the trial-by-trial correlation in each scanning session. Error bar indicates stand error of the mean across scanning sessions. (**B)** Policy sequenceness at 50 ms and 150 ms time lag, separated based on whether including "neg" states in the transitions or not. There is no significant difference between sequence strength of transitions contains vs. not contains the "neg" state.

5.2.3 Policy sequence is related to behavioral performance

If the forward policy sequence is indeed related to planning, it should also support behavior. Here I test whether the strength of neural sequenceness is related to behavioral planning performance on a trial-by-trial basis. The planning performance is defined as the rank of money earned under the current policy comparing to all possible policies at this trial. First, we found no relationship between averaged sequenceness of all valid transitions at either 150 ms or 50 ms time lag and planning performance, consistent with previous observations [53]. But interestingly, after separating policy and non-policy sequence and their directionality, I found the strength of a forward 150 ms policy sequence positively correlated with planning performance (Figure 5.6A, $p = 0.04$). In other words, the stronger the forward policy sequence is, the better the behavioral performance at the current trial, while no relationship was found for a backward 50 ms policy sequence ($p = 0.18$). Further separating the behavioral policy sequence based on whether the transitions are part of an optimal policy path, revealed this positive correlation with behavior is stronger with optimal policy path ($p = 0.007$). No effect was observed for backward replay of policy sequences at 50 ms time lag. These findings suggest a specific role for the forward 150 ms policy sequence in planning. On the other hand, the strength of backward non-policy sequence at 50 ms lag (but not 150 ms forward) is negatively correlated with planning performance (Figure 5.6A, p = 0.02), suggesting distinct behavioral functions for the 50 ms non-policy and 150 ms policy sequences.

The opposing relationship between forward policy and backward non-policy sequence during planning implies there could be neurophysiological differences between them that, of a type that map to differences in theta and SWR sequences in rodents. We have reported rodent SWR sequence like phenomenon in humans during rest in Section 3. The fast-backward sequence (with 50 ms state-to-state time lag) reported here resemble what we have found previously during rest. I hypothesised we would see a SWR power increase at the onset of a fast-backward sequence, but not at onset of a forward slower sequence. Indeed, I found SWR power increase only at the onset of 50 ms backward sequence (Figure 5.6C, p = 0.02), and not the 150 ms forward sequence (p = 0.38). In rodents, theta sequence and SWR sequence happens at macroscopically different times, and consistent with this I found an onset of fast 50 ms non-policy sequence is temporally anti-correlated with the forward 150ms policy sequence (Figure 5.6D, p = 0.002, this is also true for sequence strength, cf. Figure 5.6B). This implies that whenever a non-policy sequence happens then it is unlikely one will see a forward policy sequence. Together, the results suggest distinct neurophysiological features between fast non-policy sequence and forward policy sequence in humans, which might map to rodent SWR and theta sequences.

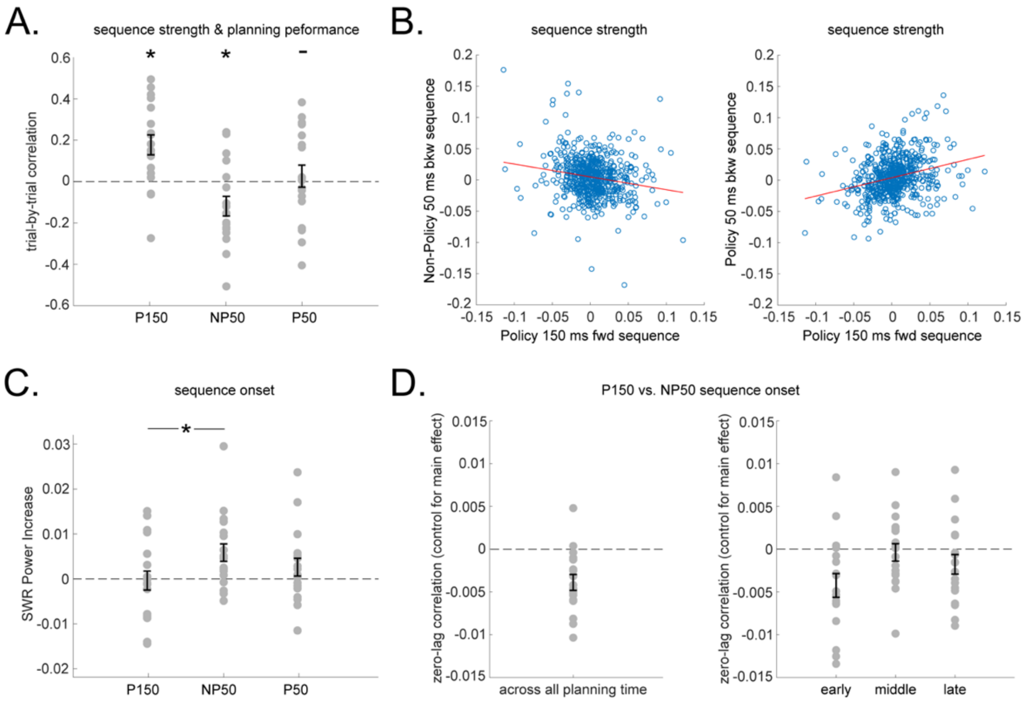


Figure 5.6 Opposing behavioral function and neurophysiological features between policy and non-policy sequences

(A) Sequence strength of a policy related transition at 150 ms (but not 50 ms) is positively related with the amount of money earnt (i.e., planning performance) on a trial-by-trial basis, while the sequence strength of non-policy (but still valid) transition at 50 ms is negatively correlated with planning performance. Here each dot indicates each scanning session. Error bar indicates stand error of the mean across scanning sessions. * indicates statistically significant, $p < 0.05$, **-** indicates non-significant. (B) Sequence strength of policy related transition at 150 ms is negatively correlated with that of non-policy transitions at 50 ms in the same trial, but positively correlated with that of policy related transitions at 50 ms. Here each scatter point is a trial, with all trials pooled over participants shown together. The red line indicates the best least-square linear fit across all trials. (**C**) Scatter plot of SWR frequency power increase at the onset of sequence events. Only 50 ms non-policy sequence event has significant SWR power increase at the sequence onset relative to sequence event time, and it is significantly higher than that of 150 ms policy sequence. Here each dot indicates each scanning session. Error bar indicates stand error of the mean across scanning sessions. * indicates statistically significant, $p<0.05$, **-** indicates non-significant. (**D**) 150 ms policy sequence event is anti-correlated with the 50 ms non-policy ones in planning time, and such anti-correlation is strongest in the early planning time within trial (the first one third of planning time).

5.2.4 Policy sequence is not a function of experience

Despite the similarity between a forward policy sequence and theta sequence in terms of neurophysiological features and relationship to behavior, it remains unclear how plans are formed in the first place. One idea is that they are related to past experience and involve replaying past experience in general. I tested the relationship between prior experience and strength of sequenceness. At each trial, for each state, I calculated when was the last time this state was visited, and the strength of sequenceness involving this state (at either 50 ms or 150 ms time lag). I found no systematic relationship between sequenceness and experience (Figure 5.7), suggesting neural sequenceness observed here is not just a replaying of past experience.

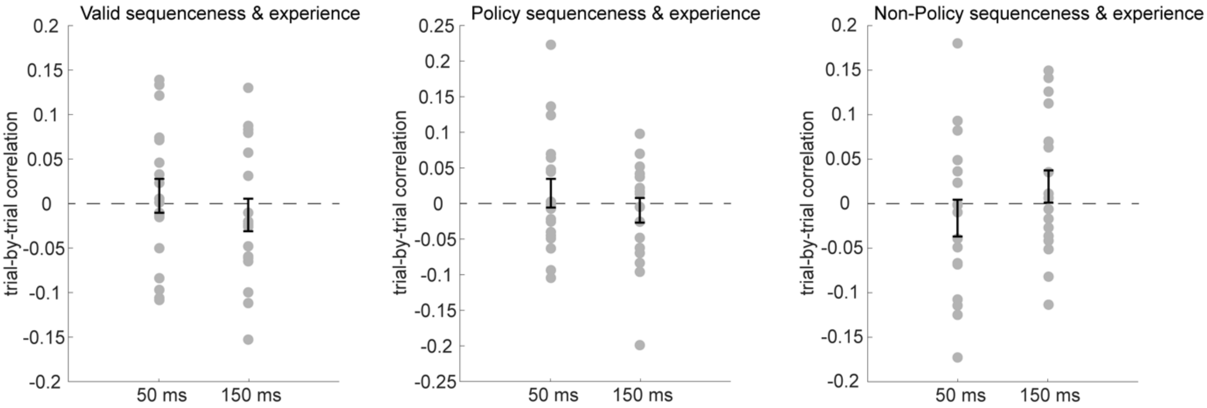


Figure 5.7 Trial-by-trial correlation between sequence strength and experience

To determine if the sequence strength is modulated by experience, at each trial, for each state, I calculated when was the last time this state is visited, and the strength of sequenceness involving this state. I do this separately for all valid transitions (left panel), transitions that are policy related (middle panel) alone and other non-policy but still valid transitions (right panel). I found no correlation between sequenceness (either at 50ms or 150ms time lag) that related to experience. Each dot indicates the trial-by-trial correlation in each scanning session. Error bar indicates stand error of the mean across scanning sessions.

5.2.5 Forming policy sequences

I observed two types of policy sequence during planning: a fast (with 50 ms state-to state time lag) backward one, and a slower (150 ms time lag) forward one. This suggests there might be a systematic relationship between the two type of policy sequences that inform how planning is formed. To test this, I first demonstrate it is methodological possible and valid to quantify temporal structure, not only within, but also between sequences in simulation (Figure 5.8A). Then, I applied the same method to quantify the gap between fast-backward policy sequence and its forward counterpart. I found the fast-backward policy sequence is temporally leading its forward slower counterpart, with a systematic gap that peaked at 240 ms time lag (Figure 5.9A, C). This temporal structure is specific to a policy sequence of the same transitions (Figure 5.9B, $p = 0.01$), and not to a non-policy sequence ($p = 0.76$) nor a policy sequence of difference transitions ($p = 0.43$). This is most evident at an early time of planning (Figure 5.8B, C). We speculate that one possible explanation might be that planning is formed by sampling from a transition model (with fast generated reverse sequences), where useful samples are incorporated into the forward sequences in a low theta frequency (240 ms ≈ 4 Hz).

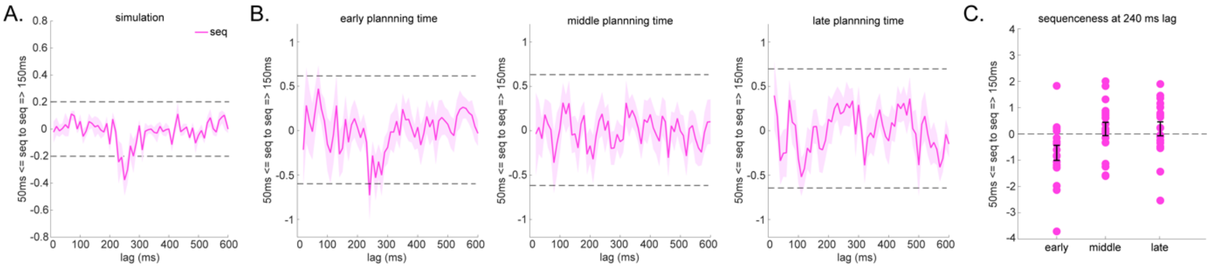


Figure 5.8 Temporal structure between the two policy sequences (simulation + breakdown)

(**A**) To validate our method, we simulated two sequences in the synthetic data (n=24), one is in a forward direction (e.g., A->B), with an optimal state-to-state time lag 150 ms; another is in a reverse direction (e.g., B->A) with an optimal state-to-state time lag 50 ms. Crucially, we specified their temporal structure in the same way as we have discovered in the real MEG data during planning, i.e., the fast reverse sequence (50 ms stare-to-state time lag) is temporally transiting to the same, but slow and forward self (150 ms stare-to-state time lag), with a gap of 240 ms between the two sequence events. Our method can successfully uncover temporal structure between sequences. The dashed line is a permutation threshold, given as the two-tail 95th percentile of the absolute value of shuffles over all lags (10-600 ms). Purple line indicates the mean sequenceness between two policy sequences across subjects. Shading indicates stand error of the mean. **(B)** Temporal structure between the two policy sequences in the early, middle and late planning time within each trial. The dashed line is the two-tail 95th percentile of the absolute value of shuffles over all lags (10-600 ms), computed separately for different portions of planning time. The only significant time lag is at 240 ms in early planning time. **(C)** Scatter plot of sequenceness between the two policy sequences at 240 ms gap, separately calculated for early, middle and late planning time. The temporal relationship is the strongest during the early time of planning.

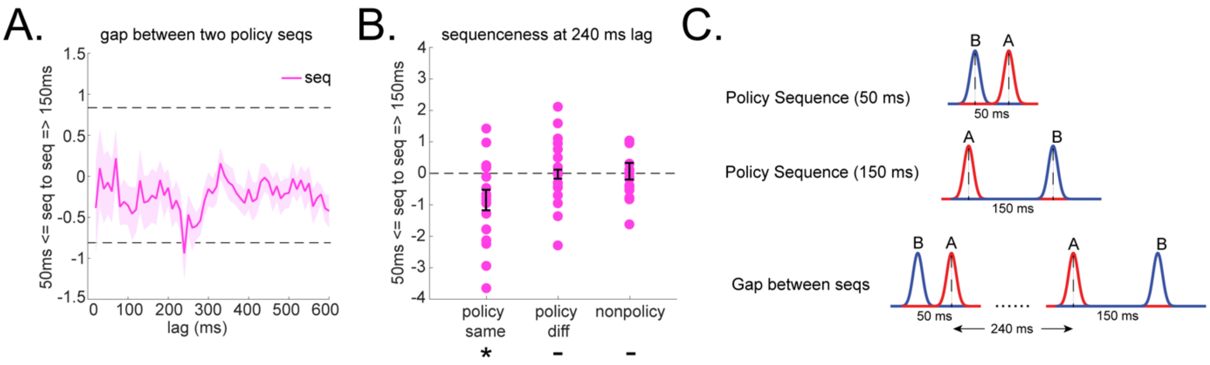


Figure 5.9 Temporal structure between two policy sequences

(**A**) Fast reverse policy sequence (50 ms state-to-state time lag) is temporally transiting to the slower and forward policy sequence (150 ms state-to-state time lag) of itself, with a gap, peaked at 240 ms between the two sequence events. Purple line indicates the mean sequenceness between two policy sequences across subjects. The dotted line is the two-tail 95th percentile of the absolute value of shuffles over all lags (10-600 ms). Shading indicates stand error of the mean. **(B**) Scatter plot of sequenceness between the two policy sequences vs. non-policy sequences at 240 ms for each scanning session. It is further separated for same vs. different transitions following the first one in the policy sequence. Such temporal structure only exists between the same transitions. **(C)** Illustration of the temporal structure of the same transition in two policy sequences.

5.3 Discussion

Utilizing the new methods advance (detailed in section 2), I am able to model forward and backward sequence direction separately, and separate policy-specific transitions from all valid transitions trial-by-trial. As a result, I expand on previous finding[53], and report the existence of two types of sequences related to planning: 50 ms backward and 150 ms forward sequences

The 50 ms backward sequence and 150ms forward sequence have distinct neurophysiological features, and an opposing behavioral function related to planning. I found the forward 150 policy sequence positively supports planning performance in a non-spatial space, while a 50 ms nonpolicy sequence is negatively correlated with behavior. The rapid 50ms sequence plays out the general statistics of problems independent of the current behavior, whereas the slower 150 ms sequence preferentially plays out the behavioral trajectory an agent is about to act out. These are potentially equivalent to theta and SWR replay as reported in rodents. It is also intriguing to note, the similarity between results in this planning project compared to project 3 on model-based learning. It is possible the same neural mechanisms underpin both model-based learning and planning, where the direction of a sequence depends on its function, e.g., reverse for value backup/credit assignment and forward for planning. I will discuss these results from all three experimental projects in the “general discussion” section and speculate on the function and relationship of human replay/sequences to that seen in rodents.

In this project, the transition between fast 50 ms backward sequence to its slower (150 ms) forward self in forming a policy sequence is particular interesting, and novel to this project. This is suggestive that possible plans are sampled using rapid sequences and a transition to the slower mode only occurs when there is an actionable plan. If this holds true, this is suggestive of a sampling-based planning model, one consistent with the general idea of planning as inference [106].

5.4 Methods

5.4.1 Participants

12 adults aged 18–31 participated in the experiment, recruited from the UCL Institute of Cognitive Neuroscience subject pool and from a mailing list for MSc students. Six were female and two were left-handed. All participants had normal or corrected-to-normal vision and had no history of psychiatric or neurological disorders. Eight of the 12 participants underwent two scanning sessions, for a total of 20 recorded sessions. Two of these sessions were excluded before the start of analysis owing to large artifacts, leaving 18 analyzed sessions. All participants provided written informed consent and consent to publish prior to start of the experiment, which was approved by the Research Ethics Committee at University College London (UK), under ethics number 1825/005.

5.4.2 Task

In the MEG scanner, participants performed a 6-state sequential reasoning task, inspired by Huys, et al. [127,128], but designed with the additional criterion of encouraging mental representation of the visual objects that identified each state. The task was implemented in MATLAB (MathWorks) using Cogent (Wellcome Trust Centre for Neuroimaging, University College London). Each trial began with participants placed at a randomly selected state within the maze. From this state, they were permitted four sequential moves with the instructed aim of maximizing their earnings. From each state, a move constituted one of two possible choices, called “up” and “down” (so-called for simplicity of button pressing, although there was no meaningful spatial relationship between the states). Each of these choices deterministically led to a different next state. Only one state was ever viewed at a time, and participants never saw an overall bird’s-eye view of the maze.

Each state provided a monetary outcome of between −5 and +5 pence. The reward for each state drifted independently at random by −1, 0, or +1 pence on each trial. Upon reaching a state, the state's current reward value was added to the participant's running total for that trial. This running total was also displayed on the screen while moves were being executed. Finally, in each trial, two randomly selected states were designated as "neg" states. When a neg state was reached, first its reward value was added to the running total for the trial as usual, but then the sign of the running total for the trial was flipped (e.g., −9 became +9 and vice versa). The identities of the neg states were signalled in text at the beginning of each trial. In many trials, the optimal strategy involved the use of one or two neg states. Two neg states could be used within a trial to reach a positive total reward, or a single neg state could be used in conjunction with negative state reward.

On each trial, participants were first shown in text the names of the starting state and the two neg states and allowed up to 60 s to plan. After the end of the planning period, participants were faced with a blank screen upon which they could pre-enter their chosen sequence of four moves. They were allowed up to 3 s to enter the first move and 1 s for each of the last three moves. As they pre-entered each move, a corresponding up or down arrow appeared on the screen for confirmation, but no visual objects were shown. After pre-entering all four moves, the visual object corresponding to the starting state of this trial appeared. Participants were then required to repeat the sequence of moves they had pre-entered. As they executed each move, the visual object shown on the screen changed to reflect the corresponding state transition. Up to 10 s was permitted to execute each move. Executing each move was followed by 350 ms of animated cross-fade transition between visual objects, followed by 500 ms pause, followed by the current reward amount of the new state displayed for 1,000 ms, followed by the total trial earnings being updated and displayed for 1,000 ms, followed by a neg and corresponding change to total trial earnings, if any, being displayed for 1,000 ms. If this was the final move of the trial, the final reward for the trial was then displayed for 3,000 ms; otherwise, the next move could be entered.

The task design was motivated by a wish to encourage participants to learn and exploit the transition structure of the task, instead of relying on simple choice strategies like repeat reinforced actions. I reasoned that engaging participants with the transition structure would afford the best chance of detecting neural sequences reflecting this structure. Two other features of the task design were also intended to meet this purpose. First, trials were generated such that simple choice strategies would yield much lower payout than optimal planning strategies. Second, pre-entry of the four sequential moves on each trial was made in the absence of feedback about the consequences of those moves until all four had been entered. This meant that participants had to anticipate where move m would lead to in order to make a good decision on move $m + 1$.

Each of the six states in the maze was a unique visual object. For each participant, the six objects were drawn randomly from a set of ten objects (bird, bread, cat, chair, garlic, hammer, hand, horn, tree, water), and the six chosen objects were randomly assigned to the six states of the maze.

After the 6-state reasoning task, participants completed a secondary task while still in the scanner. This task was designed to elicit neural representations of known stimuli, which could be used to train classification models. In this secondary task, the name of a visual object appeared in text for a variable duration of 1,500 to 3,000 ms, followed immediately by the visual object itself. On 20% of trials, the object was upside-down. To maintain attention, participants were instructed to press one button if the object was correct-side-up, and a different button if it was upside-down. Once the participant pressed a button, the object was replaced with a green fixation cross if the response was correct and a red cross if the response was incorrect. This was followed by a variable length inter-trial interval of 700 to 1,700 ms.

Each session included 125 trials of the secondary task, with approximately 16 correct side-up presentations of each visual object. Only correct-side-up presentations were used for classifier training. The trial order was randomized for each participant. Per participant, the visual objects used were the same six objects used in the main task.

5.4.3 MEG Acquisition and Pre-processing

MEG was recorded continuously at 600 samples/second using a whole-head 275-channel axial gradiometer system (CTF Omega, VSM MedTech), while participants sat upright inside the scanner. Participants made responses on three buttons (called “up,” “down,” and “advance”) of a button box using the fingers they found most comfortable.

The data were resampled from 600 Hz to 100 Hz to conserve processing time and improve signal to noise ratio. Thus, data samples used for analysis were spaced every 10 ms. All data were then high pass filtered at 0.5 Hz using a first order IIR filter to remove slow drift. All analyses were performed directly on the filtered, cleaned MEG signal, consisting of a length 134 vector of samples every 10 ms, in units of femtotesla.

5.4.4 Multivariate MEG Analysis

A multi-class lasso-regularized logistic regression models were trained on MEG data elicited by direct presentations of the visual objects. These presentations were taken from the secondary task that succeeded the 6-state reasoning task in the scanner, specifically the data 200 ms following stimulus onset. The model is trained by adding the same amount of null data (with label “0”) and treating the null data as the reference category.

We select the L1 hyperparameter that gives the highest decoding accuracy on average. The accuracy is estimated by treating the index of the model with highest probability output as the predicted object in a leave-one-out cross-validation scheme. We then used the trained model with the chosen L1 to make predictions as to whether unlabelled MEG data corresponded to a neural representation of visual object k. Each time point was treated independently. At each time point in the unlabelled data, the data vector over sensors was multiplied by classifier weights and transformed by a sigmoid to obtain a predicted probability for visual object k. This procedure yielded six probabilities at each time point (excluding the null class). For each trial, we obtained a matrix X with six columns and as many rows as time bins in the trial.

5.4.5 Sequenceness Measure

Previously, we have relied on the asymmetry between forward and backward cross correlation to account for general correlation between states. In the current method, we tried to account for general correlation between states by constructing two-level GLMs. We call this approach: "temporal delayed linear modelling" (TDLM). This is described in detail in Section 2. Briefly, at first level, all decoded states with same time lagged copies are included in the design matrix and regressed onto all raw state time series separately. This controls for covariance among states and result in a pairwise sequence matrix at each time lag. At second level GLMs, non-specific general dynamics, and auto-transitions are controlled while looking for the sequenceness of transition of interest. In sum, the two-level GLMs is asking, on average, to what extent does the state i at time lag Δt uniquely predict state j, compared to evidence of all other transitions.

5.4.6 Sequence of Sequence

The same approach is capable of quantifying not only the state-to-state transitions, but also sequence-to-sequence dynamics after a change of state space. To quantify sequence of sequence, TDLM needs to construct the design matrix to carefully control for state-to-state effects. In the linear model, this is effectively asking for the interaction effect of A and B, one should therefore control for main effect of A and B alone. Similar with quantifying state-to-state transitions, TDLM operate in two-level GLMs to measure the sequence-to-sequence transitions, but with extra control of state-to-state effects.

Let's assume the sequence state is X_{seq}, after transforming original state to sequence based on the optimal state-to-state time lag Δst. Each entry at X_{seq} is sequence state, denoted by $ij_{\Delta st}$, which means the sequence i -> j with the optimal state-to-state time lag Δst. In the first level GLM, TDLM ask the extent of unique contribution of sequence of sequence (i.e., interaction effect), while controlling for sequence of states (i.e., main effects). For each sequence state $ij_{\Delta st}$, at each possible time lag Δt, TDLM estimated a separate linear model:

$$Y_{ij_{\Delta st}} = X_i(\Delta t)\beta_i(\Delta t) + X_{j_{\Delta st}}(\Delta t)\beta_j(\Delta t) + X_{ij_{\Delta st}}(\Delta t)\beta_{ij_{\Delta st}}(\Delta t)$$

The predictors $X_{ij_{\Delta st}}(\Delta t)$ is the time-lagged copies of the sequence state $ij_{\Delta st}$ reactivation timeseries. The model predicted $Y_{ij_{\Delta st}}$, the reactivation of sequence state $ij_{\Delta st}$. The nuisance regressors are $X_i(\Delta t)$ - the time-lagged copies of the original state i; and $X_{j_{\Delta st}}(\Delta t)$ - the time-lagged copies of the lagged state j. Repeat this process for each sequence state separately at each time lag, resulting a sequence matrix $\beta_{seq}(\Delta t)$. In the 2nd level GLM, TDLM asks how strong the evidence of sequence of interest is compared to sequences that have the same starting state or end state at each time lag.

5.4.7 Statistical Test

For all statistical testing of sequenceness in this paper, we only tested if the peak sequenceness across all time lags (from 10 to 600 ms) is significantly different compared to zero. This is done by between-subject sign flip permutation test, with 5000 permutations. I did not perform the same statistical test as in my previous paper because here I am interested in the question - do we have significant sequenceness of certain transitions (e.g., policy related transitions) compared to zero, rather than is this sequenceness the strongest compared to any other transitions. There is no multiple comparison problem as only the peak point is tested against zero. For the participants who had two sessions, regression models were trained separately for each session to account for a possibility of different head position, and sequenceness measures from the two sessions were averaged together before computing group-level statistics.

5.4.8 Stepwise Policy Sequence

In addition to measure the average evidence of sequenceness for policy related transitions, we have also reported the strength of evidence for each step of policy trajectory in the same order as a behavioral readout at 50 ms (backward) and 150 ms (forward) separately (Figure 5.5). The evidence of sequenceness for each state pair of permitted transitions is shown in Supplementary Figure 5 from Kurth-Nelson, et al. [53]. We have also reported evidence of policy sequence including versus not including “neg” state, given the importance of “neg” state in this task (the accumulated earning will flip sign when encounter the “neg” state). There is no significant difference between transitions including “neg” state and ones of only non-neg (i.e., normal) states.

5.4.9 Sharp Wave Ripples (SWR) and Non-policy Sequence

In rodents, spontaneous offline replay events co-occur with bursts of high frequency (120-200 Hz) local field potential power known as sharp wave ripples (SWRs). I have found similar power increases (120-150Hz) at replay onset during rest in humans (Section 3 and 4). Here, I tested whether there is a difference in terms of SWR power between policy and non-policy sequences onset, as with previous projects (Section 3 and 4). The individual sequence events were defined as moments with high probability of a stimulus reactivation that were followed by high probability of reactivation of the next stimulus in the sequence with given time lag (50 ms for non-policy sequence, 150 ms for policy sequence). I calculated this sequence onset using a “shift and multiply” approach described above to identify the sequence time series, then thresholding it at its 95th percentile with an additional constraint that it has 100 ms of sequence-free time preceding them. This constraint ensures we identify the onset rather than middle of a sequence (if it is multi-length). To control for general difference between policy and non-policy sequence, the SWR power is demeaned within each sequence event before the comparison.

6 GENERAL DISCUSSION

6.1 Is spontaneous neural activity useful?

Based upon the evidence presented in this thesis, it seems the answer to this question is yes. I have found that internally generated neural activity – replay, is richly structured. It is not just an "echo" of past experience. In the service of efficient inference and learning in humans, I have found human replay reorganize past experience (Section 3) and prioritize non-local experience (Section 4). During decision time, human replay supports sequential multi-step planning, delineating the path a subject is about to take in a non-spatial space (Section 5).

In reinforcement learning (RL) terminology, non-local replay can be seen as neural correlate of a model-based computation [32]. The model is a mental model, describing what states are, and how they are connected to each other. This concept is similar to schema, or a situational model in psychology literature [129]. But for consistency, I will stick with RL terminology. The computation is model based, because it is unrelated to direct experience, it relies an internal model to generate novel experience (Section 3), samples a distant experience (Section 4) or plans in multi-steps (Section 5). So far, I have not considered what specific algorithm replay might correspond to functionally. In the following section, I will allow some speculations on this.

6.1.1 Building efficient representation for inference and generalization

First, there is the question of how an efficient task representation can be built through replay. This is perhaps the most difficult question for RL. RL mostly concerns about learning and control on a defined state space, but discusses little about how to build one in the first place [6]. The only requirement of state space in RL is that it has be Markovian, meaning the future state depends, and only depends, upon the present state and not on any past state. There are countless alternative state spaces that can satisfy this requirement. The question is how to build and/or select one that is best for the current task. This is most relevant for a novel environment where scarce direct experience is available. To recapitulate an example, what do we do when we land at a new airport?

One design principle of this efficient representation, as suggested from Section 3, is factorization [130]. We can factorize the representation of sensory and structural knowledge, so that the abstract structural knowledge can be learnt from the past and re-used in a novel environment [10]. When encounter a novel environment, this structural knowledge then combines with new sensory information to form a representation of the task at hand. This is efficient because we do not need to re-learn everything anew. We can generalize past experience to infer the appropriate behavior in the current task.

Our results suggest replay helps this process, in a sense that we have observed replay of both structural and sensory representation, where structural replay guides sensory replay into its correct template. But for now, we can only say replay is a neural correlate of this process and, as yet we don't know why and how it is useful.

Recent RL theory suggests replay might be explained as a prioritization of memory access that helps maximize future reward *within* an environment, by extending a one-step Bellman backup to multi-steps [34]. This theory does not help us here, however, because it cannot explain how replay helps generalization *across* environments.

How should we approach this question? One interesting idea comes from recent success in building deep neural network to explain cellular activity in the hippocampus and entorhinal cortex (the same cells where replay occurs). One type of model, termed "Tolman Eichenbaum Machine" (TEM) [131] is of particular interest. TEM learns an abstract representation of the relational structure that has similarities to cellular activity seen in entorhinal cortex, e.g., border cell, object vector cell, and the famous grid cell, when combined with sensory information, TEM can infer what will be seen next.

One can plug replay into this model, under the same objective function, and ask whether the inclusion of replay facilitate the learning process of the model. Perhaps, more interestingly, one can use a different objective function, e.g., not only maximize the predictive ability in novel contexts, but optimize the speed of doing so. This will require an abstract representation to be learnt in the most efficient way and will, in turn, pressure replay to prioritize memories that are most informative for abstracting task structure. This gives an opportunity to examine novel replay that addressed this aim, e.g., an alternative path, the one that is not taken, but can be inferred from past experience, may be preferentially replayed because "seeing" the same thing from different angles is useful for extracting the underlying structure.

6.1.2 Bridging space and time to solve credit assignment

Credit assignment has long been an important problem in RL [132] where the credit is mostly a synonym for value. The problem occurs because the RL agents try to maximize the future cumulative reward, not just the current one. In fact, in most cases where reward is sparse, the outcome might only occur at the end of a trial. For example, in the game of GO the reward is win or loss at the very end. How to assign this final outcome to earlier actions is an extreme example of a temporal credit assignment problem. This problem also occurs in the spatial domain. For example, actions at different states can have the same outcome effect, how to update value of non-local states based on the outcome received from local experience is a spatial credit assignment problem. To solve credit assignment, we need a mechanism that can link distant space and time. Replay is an ideal candidate, given it is an internally generated, sequential pattern of neural activity. The results from section 4 provide the evidence that neural replay is used to solve the credit assignment problem. Note, unlike most rodent studies [20], I did not look for replay of direct past experience but instead focus on replay of non-local experience. This is because while local learning can be explained by other mechanisms, e.g., eligibility trace[133], non-local learning must be based on a model (or a memory pool, we don't separate them here). In fact, I find this non-local replay solves credit assignment in a similar way to that predicted by a RL theory, where replay is seen to prioritise a memory access according to its utility in maximizing future reward [34].

6.1.3 Forming plans through sampling

The section 5 provides an intriguing insight into how planning is formed. I observed a fast, presumably SWR replay based, sampling process, where actionable plans were encoded in a forward slower sequence. This slower sequence had a state-to-state time lag of 150 ms, roughly 6-7Hz, within the theta frequency. It is possible this captures a theta-like sequence in rodents. Although we do not have further evidence to support this idea, it remains a speculation.

This sampling account was not an original hypothesis. Originally, I thought of human planning as akin to tree search [134], which has been suggested from behavioral studies (with certain heuristics, e.g., "pruning" [127], "plan-until-habit"[135]), and shown to be an effective algorithm in solving complicated planning problems in deep neural network models, e.g., GO [12]. Instead, I found no evidence to support this. For example, one would expect a structured sequence to sequence transitions in behavior trajectory, e.g., A->B, leads to B->C, in a systematic way, if tree search like computation is true. Although one might argue the fact, we see replay strength correlates with behavioral performance could be a sign of Monte Carlo Tree Search (MCTS) like computation, i.e., transitions are favoured in model-based reasoning to the extent that they are good. This requires more evidence.

It is interesting to speculate that the brain chooses to use a different implementation for a good reason. One idea is that sampling is easier and a relatively simple computation to implement. Recent theoretical work suggests fast sampling may be a universal computation that the brain implements for inference [136]. This remains a speculative guess as to the role of replay and an idea that needs to be thoroughly tested in future.

6.2 What is replay in humans?

The question here relates to the physiology of spontaneous neural activity in humans. For simplicity, I have named all spontaneously generated sequences of neural reactivations - replay. I consider it to be internally generated, i.e., not caused by current sensory input, and sequential activity, not mere reactivation.

But is this equivalent to rodent replay? I have tried to answer this in section 3. Across two studies I have shown these spontaneous sequences of cortical events, as detected in human MEG recordings, in a non-spatial space have strong parallels to hippocampal replay observed in rodents during sharp-wave ripple epochs in spatial tasks. Like rodent replay (i) they appear spontaneously during rest, (ii) they compress time from seconds to tens of milliseconds, (iii) they reverse in direction after reward and (iv) they involve coordination between hippocampus and sensory cortex, and (v) they are associated with power increase in ripple frequency (120 Hz -150 Hz), source localized to hippocampus. I have also replicated these same findings in section 4 and 5. Furthermore, in a recent study, with colleagues I have applied the hidden markov model to the same resting state data to infer the dynamics of resting state networks [137], and shown that such sequence events temporally correspond to the activation of the default mode network. This work provides a powerful linkage between human replay and whole brain network activity [137].

In addition to this fast 20-50 ms SWR like replay, I have also identified a slower type of replay, with a time lag of 150 – 160 ms, as detailed in both section 4 and 5 on model-based learning and planning. What does this signature correspond to? The short answer is we, as yet don't know. I have speculated it might relate to theta sequence, given its speed is roughly in a theta frequency, its onset is not associated with a power increase in ripple frequency and its strength is related to an on-task computation, e.g., model-based learning (section 4), and sequential decision-making (section 5). But it is also possible that this 150ms (ish) human replay does not have rodent equivalent either due to the species related differences or due to the fact that MEG is measuring neural activity at population level, rather than measuring activity in cell ensembles.

In addition to speed, the direction of replay remains an intriguing subject that deserves more work and attention. In rodent literature, SWR replay during rest can be either forward or reverse[120], where reverse replay seems to be uniquely modulated by reward [20]. This contrasts with a theta sequence which is dominantly forward, akin to a look ahead signal [42,45,66]. We found similar signatures in human replay, that it can be either forward or reverse, including a flipping of directions after reward. But the picture becomes complex when we look at the slower speed (150-160 ms) replay, which is reverse for model-based reward learning and forward for planning. It is intriguing to speculate that the direction of replay might be functional meaningful and can be modulated by task demand. For example, the 150-160 ms replay plays out in a backward direction during outcome receipt because it need backpropagate the prediction error signal from the final state to the initial state. The same replay goes in the forward direction during decision time because it needs to think from local (i.e., state state) to the final states (e.g., in section 5). Relatedly, in one of our recent studies, we have shown by simply changing the probe question shifts the direction of replay in memory retrieval [117].

In sum, I have found ripple related replay in humans. It bears a lot of interesting properties as in rodent replay. There is also suggestive evidence of theta-like sequences in humans, especially during active planning or flexible model-based learning. Unlike theta sequence in rodents which is dominantly forward (but see Wang, et al. [138]), and encodes local information, the theta-like human sequences I have identified in this thesis can go either forward (in planning) or backward (in value leaning), and it encodes mostly non-local information. This could be due to the reported difference in theta oscillation (both in term of frequency range and continuity) between human and rodents [49].

In future work, it will be interesting to delineate this picture in more detail and clarity. Notably, the method (TDLM) we have developed is a general sequence detection method that can be used on different data modalities (e.g., MEG or electrophysiology), and on any graph, rather than a one-D space map. I hope this method will help bridge results between humans and rodent research on this topic.

6.3 A new paradigm

As a final statement I would like to address the underlying philosophy behind the line of research I have presented.

In my mind, the outlined research is possible because of two changes emerging in the field. One of these is the emergence of a new conceptual paradigm. It is hard to think about studying spontaneous neural activity within the framework of an information processing metaphor, where cognition is considered a responsive process. There are studies looking at internal oriented cognition, especially on episodic memory, e.g., imagination [139], consolidation [85], though they rarely relate to on-task cognition and behavior. It remains unclear how internal and external oriented cognition are related, and what goal they are serving.

In this thesis, adopting views from RL, I consider spontaneous neural activity as a means to realize model-based computation. In this view, external and internal oriented cognition serve the same goal, i.e., they enable adaptive behavior. This idea has rich links to the idea of a “cognitive map” – as addressed in research on hippocampus in both humans and rodents. A cognitive map can be conceptualised as a model in RL terms, describing how elements or states are linked to each other, i.e., $T = p(s'|s, a)$. In the neuroscience literature, the study of episodic memory and spatial navigation (where the cognitive map concept is mostly used) rarely makes links with the terminology of RL [140,141]. It is interesting therefore that in recent realizations of model based RL, two separate domains of enquiry in cognitive neuroscience: decision-making (dopaminergic system, e.g., ventral tegmental area), and episodic memory (e.g., hippocampal-entorhinal system) have been unified [77,142]. I consider that this makes much sense for the model based RL framework outlined in this thesis.

In a model based RL framework, it is easy to appreciate the importance of studying spontaneous neural activity (or internal oriented cognition) when it comes to processing non-local information. But how to achieve this is the hard question. By definition, spontaneous neural activity need not be tied to current stimuli. We can study the rich dynamics of the spontaneous neural activity, which has been an active research field, with links to resting state networks. But this approach renders it hard to make links with task related cognition [143]. To link spontaneous neural activity and task -related cognition, we need to be able to decipher the representational content of neural activity.

This brings us to the second change, which pertains to experimental paradigm. The TDLM method is designed to first decode the representation of spontaneous activity, and then study the regularity of patterns during reactivation. This is consistent with a broader paradigm shift in recent years [31,65,83,88], from "representation-less" to the "representation-rich". In the past, we have studied decision-making in relatively abstract terms, for example, by invoking quantities such as decision variables, value or utility. To examine their neural correlates, we typically perform a "model-based" analysis where we first derive the value of these decision variables based on computational models, and then correlate a variation in these variables with fluctuations in neural activity [144]. This is an entirely appropriate strategy if we only care about mapping decision variables on to the brain. But this type of analysis will tell us little about the computational processes.

To understand the computational processes, we need to track the representation of objects (e.g., decision variables) over time. To do this, we need to first decode what "constitutes" the relevant variable in the brain. This is especially true when studying computation of non-local information, the focus of this thesis. To do so I, and others, devote a separate task to obtain a neural representation of these abstract constructs[54,65,83,117]. We call this a "representation rich" paradigm, because we are looking for neural patterns evoked by these representations. This "representation rich" paradigm is the methodological reason why we can study spontaneous neural activity - we can transform the measured dynamics of neural activity to patterns of reactivations of task-related variables.

These two sources of change, conceptual and experimental, also remind us of the importance of the language of a discipline. This speaks to a recent debate in neuroscience and psychology [145,146], which asks whether neuroscience research is served best by a dependence on psychology terms. Buzsáki [146] has argued that neuroscience needs to build its own vocabulary based on brain mechanisms, i.e., physiology, an approach termed "inside-out". Others, such as Poeppel and Adolfi [145], think the opposite holds true and that neuroscience needs psychology. In my view, this relates to a wider debate regarding which level of description should be the first focus of neuroscience. Borrowing concepts from Marr and Vaina [147], there are arguably three potential levels of analysis: 1) computation: what problem is a system trying to solve 2) algorithmic: how does it do in terms of computation; and 3) implementational: how this his realized physically in the brain. While Buzsáki [146] favours an implementational-first strategy Poeppel and Adolfi [145] advocate a computational guided approach.

I argue for a more middle ground approach. I think neuroscience research should be computation guided, but importantly it must also provide a concrete hypothesis on the nature and realisation of such computation i.e., a specification of the algorithm as well as its physical realization, i.e., implementation. In the domain of learning and decision-making, RL provides a powerful framework. It covers all three levels of analysis and, more importantly, it enables us to express an hypothesis in formal mathematical language, which leaves little room for ambiguity inherent in folk psychology, a main criticism of Buzsáki [146]. An analogy can be made with physics. Arguably, it is only with Isaac Newton, who described physical phenomena in precise mathematic language, that physics became a rigorous science. An interesting observation in decision-making research is that in grounding in RL term, Dolan and Dayan [148] have successfully predicted that the next generation of enquiry should relate to the realization of model-based computation in the brain, 7 years ago.

Undoubtedly, as a theory, RL is limited. For example, the definition of reward is hard to define in real life (e.g., wealth? well-being? or social status?). But as a formal theory, it makes both its predictions and limitations precise and clear. I consider it as a good starting point in forming a unified language for the field.

REFERENCES

1 Simon, H. A. Information processing models of cognition. *Annual review of psychology* **30**, 363-396 (1979).

2 Kimble, G. A. Behaviorism and unity in psychology. *Current Directions in Psychological Science* **9**, 208-212 (2000).

3 Raichle, M. E. *et al*. A default mode of brain function. *Proceedings of the National Academy of Sciences* **98**, 676-682 (2001).

4 Haynes, J.-D. & Rees, G. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience* **7**, 523-534 (2006).

5 Burke, J. F. *et al.* Theta and high-frequency activity mark spontaneous recall of episodic memories. *Journal of Neuroscience* **34**, 11355-11365 (2014).

6 Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*. (MIT press, 2018).

7 Botvinick, M., Wang, J. X., Dabney, W., Miller, K. J. & Kurth-Nelson, Z. Deep Reinforcement Learning and Its Neuroscientific Implications. *Neuron* (2020).

8 Siegel, S. & Allan, L. G. The widespread influence of the Rescorla-Wagner model. *Psychonomic Bulletin & Review* **3**, 314-321 (1996).

9 Sutton, R. S. & Barto, A. G. in *Proceedings of the ninth annual conference of the cognitive science society.* 355-378 (Seattle, WA).

10 Behrens, T. E. J. *et al.* What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. *Neuron* **100**, 490-509 (2018).

11 Watkins, C. J. & Dayan, P. Q-learning. *Machine learning* **8**, 279-292 (1992).

12 Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484 (2016).

13 Foster, D. J. & Wilson, M. A. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* **440**, 680 (2006).

14 Skaggs, W. E. & McNaughton, B. L. Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science* **271**, 1870-1873 (1996).

15 Wilson, M. A. & McNaughton, B. L. Reactivation of hippocampal ensemble memories during sleep. *Science* **265**, 676-679 (1994).

16 Foster, D. J. Replay comes of age. *Annual Review of Neuroscience* **40**, 581-602 (2017).

17 Pfeiffer, B. E. & Foster, D. J. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature* **497**, 74 (2013).

18 Dragoi, G. & Tonegawa, S. Preplay of future place cell sequences by hippocampal cellular assemblies. *Nature* **469**, 397 (2011).

19 Davidson, T. J., Kloosterman, F. & Wilson, M. A. Hippocampal replay of extended experience. *Neuron* **63**, 497-507 (2009).

20 Ambrose, R. E., Pfeiffer, B. E. & Foster, D. J. Reverse replay of hippocampal place cells is uniquely modulated by changing reward. *Neuron* **91**, 1124-1136 (2016).

21 Kaefer, K., Nardin, M., Blahna, K. & Csicsvari, J. Replay of behavioral sequences in the medial prefrontal cortex during rule switching. *Neuron* (2020).

22 Carr, M. F., Jadhav, S. P. & Frank, L. M. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience* **14**, 147 (2011).

23 Ólafsdóttir, H. F., Barry, C., Saleem, A. B., Hassabis, D. & Spiers, H. J. Hippocampal place cells construct reward related sequences through unexplored space. *eLife* **4**, e06063 (2015).

24 O’Neill, J., Boccara, C. N., Stella, F., Schoenenberger, P. & Csicsvari, J. Superficial layers of the medial entorhinal cortex replay independently of the hippocampus. *Science* **355**, 184-188 (2017).

25 Ji, D. & Wilson, M. A. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience* **10**, 100 (2007).

26 Ólafsdóttir, H. F., Carpenter, F. & Barry, C. Coordinated grid and place cell replay during rest. *Nature neuroscience* **19**, 792 (2016).

27 Ólafsdóttir, H. F., Carpenter, F. & Barry, C. Task Demands Predict a Dynamic Switch in the Content of Awake Hippocampal Replay. *Neuron* **96**, 925-935.e926 (2017).

28 Vaz, A. P., Wittig, J. H., Inati, S. K. & Zaghloul, K. A. Replay of cortical spiking sequences during human memory retrieval. *Science* **367**, 1131-1134 (2020).

29 Tambini, A., Ketz, N. & Davachi, L. Enhanced brain correlations during rest are related to memory for recent experiences. *Neuron* **65**, 280-290 (2010).

30 Tambini, A. & D’Esposito, M. Causal Contribution of Awake Post-encoding Processes to Episodic Memory Consolidation. *Current Biology* (2020).

31 Schuck, N. W. & Niv, Y. Sequential replay of nonspatial task states in the human hippocampus. *Science* **364**, eaaw5181 (2019).

32 Sutton, R. S. Dyna, an integrated architecture for learning, planning, and reacting. **2**, 160-163 (1991).

33 Sutton, R. S. in *Machine Learning Proceedings* 216-224 (Elsevier, 1990).

34 Mattar, M. G. & Daw, N. D. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience* **21**, 1609 (2018).

35 Tolman, E. C. Cognitive maps in rats and men. *Psychological review* **55**, 189 (1948).

36 Gupta, A. S., van der Meer, M. A., Touretzky, D. S. & Redish, A. D. Hippocampal replay is not a simple function of experience. *Neuron* **65**, 695-705 (2010).

37 Constantinescu, A. O., O’Reilly, J. X. & Behrens, T. E. J. Organizing conceptual knowledge in humans with a gridlike code. *Science* **352**, 1464 (2016).

38 Doeller, C. F., Barry, C. & Burgess, N. Evidence for grid cells in a human memory network. *Nature* **463**, 657-661 (2010).

39 McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I. & Moser, M.-B. Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience* **7**, 663 (2006).

40 Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells, and the brain's spatial representation system. *Annual Review of Neuroscience* **31** (2008).

41 Bellmund, J. L., Gärdenfors, P., Moser, E. I. & Doeller, C. F. Navigating cognition: Spatial codes for human thinking. *Science* **362**, eaat6766 (2018).

42 Buzsáki, G. & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience* **16**, 130 (2013).

43 Dehaene, S., Meyniel, F., Wacongne, C., Wang, L. & Pallier, C. The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron* **88**, 2-19 (2015).

44 Eichenbaum, H. Prefrontal–hippocampal interactions in episodic memory. *Nature Reviews Neuroscience* **18**, 547 (2017).

45 Buzsáki, G. Theta oscillations in the hippocampus. *Neuron* **33**, 325-340 (2002).

46 Skaggs, W. E., McNaughton, B. L., Wilson, M. A. & Barnes, C. A. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* **6**, 149-172 (1996).

47 Heusser, A. C., Poeppel, D., Ezzyat, Y. & Davachi, L. Episodic sequence memory is supported by a theta–gamma phase code. *Nature neuroscience* **19**, 1374-1380 (2016).

48 Kahana, M. J., Sekuler, R., Caplan, J. B., Kirschen, M. & Madsen, J. R. Human theta oscillations exhibit task dependence during virtual maze navigation. *Nature* **399**, 781-784 (1999).

49 Herweg, N. A., Solomon, E. A. & Kahana, M. J. Theta oscillations in human memory. *Trends in Cognitive Sciences* **24**, 208-227 (2020).

50 Haxby, J. V., Connolly, A. C. & Guntupalli, J. S. Decoding neural representational spaces using multivariate pattern analysis. *Annual review of neuroscience* **37**, 435-456 (2014).

51 Kriegeskorte, N., Mur, M. & Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience* **2**, 4 (2008).

52 Biswal, B. B. *et al.* Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences* **107**, 4734-4739 (2010).

53 Kurth-Nelson, Z., Economides, M., Dolan, Raymond J. & Dayan, P. Fast Sequences of Non-spatial State Representations in Humans. *Neuron* **91**, 194-204 (2016).

54 Liu, Y., Dolan, R. J., Kurth-Nelson, Z. & Behrens, T. E. J. Human replay spontaneously reorganizes experience. *Cell* **178**, 640-652 (2019).

55 Mehta, M., Lee, A. & Wilson, M. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature* **417**, 741 (2002).

56 Grosmark, A. D. & Buzsáki, G. Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences. *Science* **351**, 1440-1443 (2016).

57 Kurth-Nelson, Z., Barnes, G., Sejdinovic, D., Dolan, R. & Dayan, P. Temporal structure in associative retrieval. *eLife* **4**, e04919 (2015).

58 Wimmer, G. E., Liu, Y., Vehar, N., Behrens, T. E. & Dolan, R. J. Episodic memory retrieval is supported by rapid replay of episode content. *bioRxiv*, 758185 (2019).

59 Fyhn, M., Hafting, T., Treves, A., Moser, M.-B. & Moser, E. I. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature* **446**, 190 (2007).

60 Sun, C., Yang, W., Martin, J. & Tonegawa, S. Hippocampal neurons represent events as transferable units of experience. *Nature Neuroscience*, 1-13 (2020).

61 Kobak, D. *et al.* Demixed principal component analysis of neural population data. *Elife* **5**, e10989 (2016).

62 Colclough, G. L., Brookes, M. J., Smith, S. M. & Woolrich, M. W. A symmetric multivariate leakage correction for MEG connectomes. *Neuroimage* **117**, 439-448 (2015).

63 Deodatis, G. & Shinozuka, M. Auto-regressive model for nonstationary stochastic processes. *Journal of engineering mechanics* **114**, 1995-2012 (1988).

64 Eichler, M. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics* **137**, 334-353 (2007).

65 Eldar, E., Bae, G. J., Kurth-Nelson, Z., Dayan, P. & Dolan, R. J. Magnetoencephalography decoding reveals structural differences within integrative decision processes. *Nature human behaviour* **2**, 670-681 (2018).

66 Lubenov, E. V. & Siapas, A. G. Hippocampal theta oscillations are travelling waves. *Nature* **459**, 534-539 (2009).

67 Wilson, H. R., Blake, R. & Lee, S.-H. Dynamics of travelling waves in visual perception. *Nature* **412**, 907-910 (2001).

68 Weinberger, K. Q., Blitzer, J. & Saul, L. K. in *Advances in neural information processing systems.* 1473-1480.

69 Higgins, C. *Uncovering temporal structure in neural data with statistical machine learning models*, University of Oxford, (2019).

70 Vidaurre, D., Smith, S. M. & Woolrich, M. W. Brain network dynamics are hierarchically organized in time. *Proceedings of the National Academy of Sciences* **114**, 12827-12832 (2017).

71 Dragoi, G. & Tonegawa, S. Selection of preconfigured cell assemblies for representation of novel spatial experiences. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **369**, 20120522 (2014).

72 Worsley, K. J. *et al.* A unified statistical approach for determining significant signals in images of cerebral activation. *Human brain mapping* **4**, 58-73 (1996).

73 Nichols, T. E. Multiple testing corrections, nonparametric methods, and random field theory. *Neuroimage* **62**, 811-815 (2012).

74 Messinger, A., Squire, L. R., Zola, S. M. & Albright, T. D. Neuronal representations of stimulus associations develop in the temporal lobe during learning. *Proceedings of the National Academy of Sciences* **98**, 12239-12244 (2001).

75 Sakai, K. & Miyashita, Y. Neural organization for the long-term memory of paired associates. *Nature* **354**, 152-155 (1991).

76 Barron, H. C., Dolan, R. J. & Behrens, T. E. Online evaluation of novel choices by simultaneous representation of multiple memories. *Nature neuroscience* **16**, 1492 (2013).

77 Wimmer, G. E. & Shohamy, D. Preference by association: how memory mechanisms in the hippocampus bias decisions. *Science* **338**, 270-273 (2012).

78 Schapiro, A. C., Rogers, T. T., Cordova, N. I., Turk-Browne, N. B. & Botvinick, M. M. Neural representations of events arise from temporal community structure. *Nature neuroscience* **16**, 486 (2013).

79 Garvert, M. M., Dolan, R. J. & Behrens, T. E. A map of abstract relational knowledge in the human hippocampal–entorhinal cortex. *Elife* **6**, e17086 (2017).

80 Penny, W. D., Zeidman, P. & Burgess, N. Forward and backward inference in spatial cognition. *PLoS computational biology* **9** (2013).

81 Sirota, A. *et al.* Entrainment of neocortical neurons and gamma oscillations by the hippocampal theta rhythm. *Neuron* **60**, 683-697 (2008).

82 Buzsáki, G. & Vanderwolf, C. H. Cellular bases of hippocampal EEG in the behaving rat. *Brain Research Reviews* **6**, 139-171 (1983).

83 Eldar, E., Lièvre, G., Dayan, P. & Dolan, R. J. The roles of online and offline replay in planning. *BioRxiv* (2020).

84 Baker, A. P. *et al.* Fast transient networks in spontaneous human brain activity. *Elife* **3**, e01867 (2014).

85 Tambini, A. & Davachi, L. Awake Reactivation of Prior Experiences Consolidates Memories and Biases Cognition. *Trends in cognitive sciences* (2019).

86 Norman, K. A., Polyn, S. M., Detre, G. J. & Haxby, J. V. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in cognitive sciences* **10**, 424-430 (2006).

87 Lewis, P. A. & Durrant, S. J. Overlapping memory replay during sleep builds cognitive schemata. *Trends in cognitive sciences* **15**, 343-351 (2011).

88 Schuck, N. W., Cai, M. B., Wilson, R. C. & Niv, Y. Human orbitofrontal cortex represents a cognitive map of state space. *Neuron* **91**, 1402-1412 (2016).

89 Dayan, P. & Daw, N. D. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience* **8**, 429-453 (2008).

90 Wittkuhn, L. & Schuck, N. W. Faster than thought: Detecting sub-second activation sequences with sequential fMRI pattern analysis. *bioRxiv* (2020).

91 Maboudi, K. *et al.* Uncovering temporal structure in hippocampal output patterns. *ELife* **7**, e34467 (2018).

92 Van Veen, B. D., Van Drongelen, W., Yuchtman, M. & Suzuki, A. Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. *IEEE Transactions on biomedical engineering* **44**, 867-880 (1997).

93 Zhang, K., Ginzburg, I., McNaughton, B. L. & Sejnowski, T. J. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *Journal of neurophysiology* **79**, 1017-1044 (1998).

94 Tsividis, P. A., Pouncy, T., Xu, J. L., Tenenbaum, J. B. & Gershman, S. J. Human learning in Atari. (2017).

95 Pezzulo, G., van der Meer, M. A. A., Lansink, C. S. & Pennartz, C. M. A. Internally generated sequences in learning and executing goal-directed behavior. *Trends in cognitive sciences* **18**, 647-657 (2014).

96 Karlsson, M. P. & Frank, L. M. Awake replay of remote experiences in the hippocampus. *Nature neuroscience* **12**, 913 (2009).

97 Hsieh, L.-T., Gruber, M. J., Jenkins, L. J. & Ranganath, C. Hippocampal activity patterns carry information about objects in temporal context. *Neuron* **81**, 1165-1178 (2014).

98 Buzsáki, G. Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus* **25**, 1073-1188 (2015).

99 Dalal, S. *et al.* Simultaneous MEG-intracranial EEG: new insights into the ability of MEG to capture oscillatory modulations in the neocortex and the hippocampus. *Epilepsy and Behavior*, 10-1016 (2013).

100 Hillebrand, A. & Barnes, G. A quantitative assessment of the sensitivity of whole-head MEG to activity in the adult human cortex. *Neuroimage* **16**, 638-650 (2002).

101 Harlow, H. F. The formation of learning sets. *Psychological review* **56**, 51 (1949).

102 Shin, H., Lee, J. K., Kim, J. & Kim, J. in *Advances in Neural Information Processing Systems.* 2990-2999.

103 Wang, J. X. *et al.* Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience* **21**, 860 (2018).

104 Bernardi, S. *et al.* The geometry of abstraction in hippocampus and prefrontal cortex. *bioRxiv* (2018).

105 Higgins, I. *et al.* Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475* (2017).

106 Botvinick, M. & Toussaint, M. Planning as inference. *Trends in cognitive sciences* **16**, 485-488 (2012).

107 Jensen, O., Gips, B., Bergmann, T. O. & Bonnefond, M. Temporal coding organized by coupled alpha and gamma oscillations prioritize visual processing. *Trends in neurosciences* **37**, 357-369 (2014).

108 Kumaran, D. & McClelland, J. L. Generalization through the recurrent interaction of episodic memories: a model of the hippocampal system. *Psychological review* **119**, 573 (2012).

109 Girardeau, G., Benchenane, K., Wiener, S. I., Buzsáki, G. & Zugaro, M. B. Selective suppression of hippocampal ripples impairs spatial memory. *Nature neuroscience* **12**, 1222 (2009).

110 Jadhav, S. P., Kemere, C., German, P. W. & Frank, L. M. Awake hippocampal sharp-wave ripples support spatial memory. *Science* **336**, 1454-1458 (2012).

111 Kornysheva, K. *et al.* Neural competitive queuing of ordinal structure underlies skilled sequential action. *Neuron* (2019).

112 Liu, K., Sibille, J. & Dragoi, G. Generative Predictive Codes by Multiplexed Hippocampal Neuronal Tuplets. *Neuron* **99**, 1329-1341 (2018).

113 Luyckx, F., Nili, H., Spitzer, B. & Summerfield, C. Neural structure mapping in human probabilistic reward learning. *Elife* **8**, e42816 (2019).

114 Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275**, 1593-1599 (1997).

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

116 Liu, Y., Dolan, R., Penagos-Vargas, H. L., Kurth-Nelson, Z. & Behrens, T. E. Measuring Sequences of Representations with Temporally Delayed Linear Modelling. *bioRxiv* (2020).

117 Wimmer, G. E., Liu, Y., Vehar, N., Behrens, T. E. J. & Dolan, R. J. Episodic memory retrieval success is associated with rapid replay of episode content. *Nature Neuroscience* (2020).

118 Doll, B. B., Duncan, K. D., Simon, D. A., Shohamy, D. & Daw, N. D. Model-based choices involve prospective neural activity. *Nature neuroscience* **18**, 767 (2015).

119 Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P. & Dolan, R. J. Model-based influences on humans' choices and striatal prediction errors. *Neuron* **69**, 1204-1215 (2011).

120 Diba, K. & Buzsáki, G. Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience* **10**, 1241 (2007).

121 Momennejad, I., Otto, A. R., Daw, N. D. & Norman, K. A. Offline replay supports planning in human reinforcement learning. *Elife* **7**, e32548 (2018).

122 Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS computational biology* **13**, e1005768 (2017).

123 Vikbladh, O. M. *et al.* Hippocampal contributions to model-based planning and spatial memory. *Neuron* **102**, 683-693. e684 (2019).

124 Gelman, A. & Rubin, D. B. Inference from iterative simulation using multiple sequences. *Statistical science* **7**, 457-472 (1992).

125 Gelman, A. *et al. Bayesian data analysis*. (CRC press, 2013).

126 Redish, A. D. Vicarious trial and error. *Nature Reviews Neuroscience* **17**, 147 (2016).

127 Huys, Q. J. *et al.* Bonsai trees in your head: how the Pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS computational biology* **8**, e1002410 (2012).

128 Huys, Q. J. *et al.* Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences* **112**, 3098-3103 (2015).

129 Ranganath, C. & Ritchey, M. Two cortical systems for memory-guided behaviour. *Nature Reviews Neuroscience* **13**, 713-726 (2012).

130 Kim, H. & Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983* (2018).

131 Whittington, J. C. *et al.* The Tolman-Eichenbaum Machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv*, 770495 (2019).

132 Sutton, R. S. Temporal Credit Assignment in Reinforcement Learning. (1985).

133 Singh, S. P. & Sutton, R. S. Reinforcement learning with replacing eligibility traces. *Machine learning* **22**, 123-158 (1996).

134 Coulom, R. in *International conference on computers and games.* 72-83 (Springer).

135 Keramati, M., Smittenaar, P., Dolan, R. J. & Dayan, P. Adaptive integration of habits into depth-limited planning defines a habitual-goal–directed spectrum. *Proceedings of the National Academy of Sciences* **113**, 12868-12873 (2016).

136 Echeveste, R., Aitchison, L., Hennequin, G. & Lengyel, M. Cortical-like dynamics in recurrent circuits optimized for sampling-based probabilistic inference. *bioRxiv*, 696088 (2020).

137 Higgins, C. *et al.* Replay bursts coincide with activation of the default mode and parietal alpha network. *bioRxiv* (2020).

138 Wang, M., Foster, D. J. & Pfeiffer, B. E. Alternating sequences of future and past behavior encoded within hippocampal theta oscillations. *Science* **370**, 247, doi:10.1126/science.abb4151 (2020).

139 Hassabis, D., Kumaran, D., Vann, S. D. & Maguire, E. A. Patients with hippocampal amnesia cannot imagine new experiences. *Proceedings of the National Academy of Sciences* **104**, 1726-1731 (2007).

140 Lisman, J. *et al.* Viewpoints: how the hippocampus contributes to memory, navigation and cognition. *Nature neuroscience* **20**, 1434-1447 (2017).

141 Epstein, R. A., Patai, E. Z., Julian, J. B. & Spiers, H. J. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience* **20**, 1504 (2017).

142 Shohamy, D. & Daw, N. D. Integrating memories to guide decisions. *Current Opinion in Behavioral Sciences* **5**, 85-90 (2015).

143 Mantini, D., Perrucci, M. G., Del Gratta, C., Romani, G. L. & Corbetta, M. Electrophysiological signatures of resting state networks in the human brain. *Proceedings of the National Academy of Sciences* **104**, 13170-13175 (2007).

144 O'DOHERTY, J. P., Hampton, A. & Kim, H. Model-based fMRI and its application to reward learning and decision making. *Annals of the New York Academy of sciences* **1104**, 35-53 (2007).

145 Poeppel, D. & Adolfi, F. Against the Epistemological Primacy of the Hardware: The Brain from Inside Out, Turned Upside Down. *eNeuro* **7** (2020).

146 Buzsáki, G. The Brain–Cognitive Behavior Problem: A Retrospective. *eNeuro* **7** (2020).

147 Marr, D. & Vaina, L. Representation and recognition of the movements of shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **214**, 501-524 (1982).

148 Dolan, R. J. & Dayan, P. Goals and habits in the brain. *Neuron* **80**, 312-325 (2013).

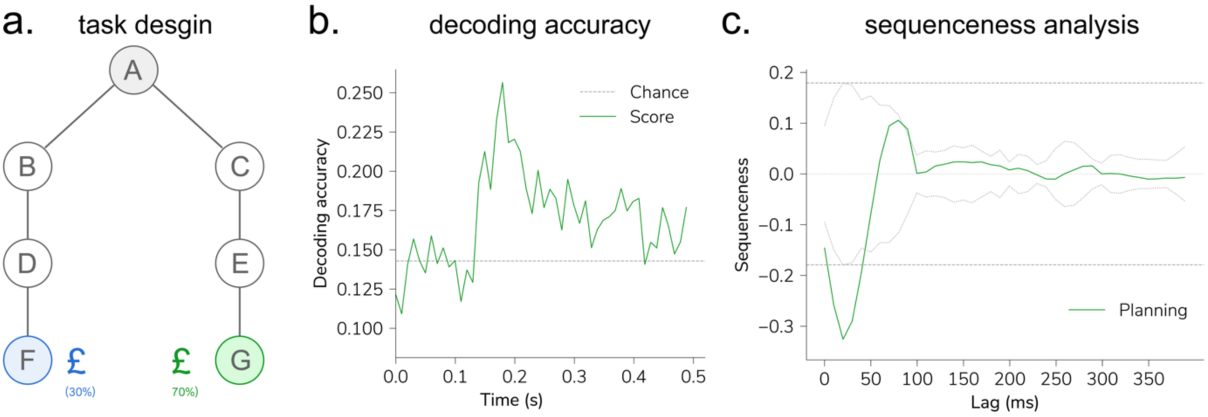
APPENDICES

APPENDIX 1: Apply TDLM to human whole-brain EEG data

TDLM is developed with autocorrelation in mind. The autocorrelation is a common place in neuroimaging data, including EEG and fMRI. TDLM approach is designed to specifically taking care of this confound and should be able to work directly with EEG and fMRI data.

We do not have the suitable fMRI data at hand to test TDLM, it seems it is better to work on the data within certain frequency range (cf., Wittkuhn & Schuck 2020). We are interested to investigate this in more depth in future work.

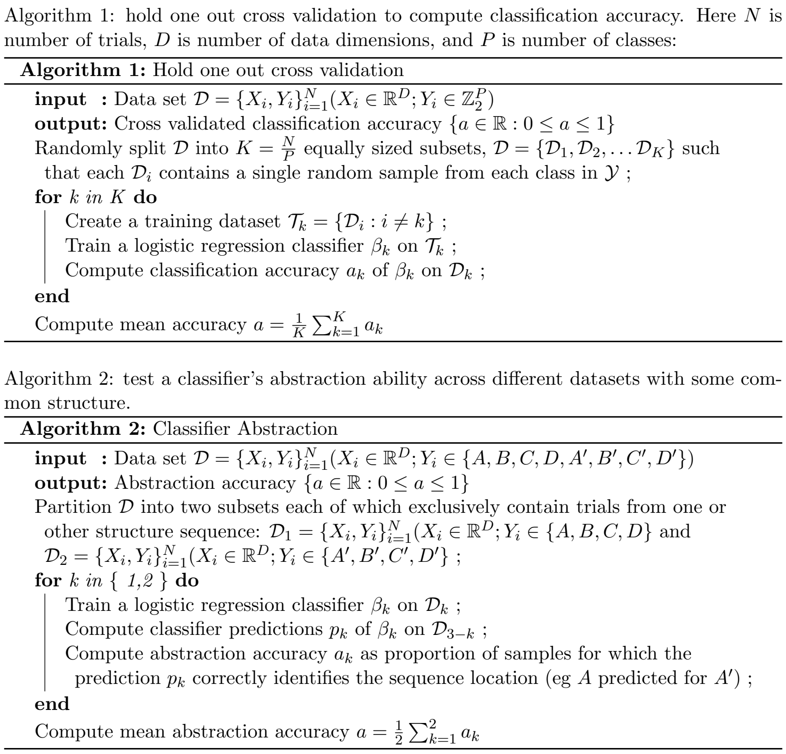
We did have collected EEG data from one participant to test whether TDLM would *just* work. The task is designed to look for online sequential replay in a decision-making task by Toby Wise. It is a 'T-maze" like task, the participant needs to choose left or right based on the value received in the end of the path. We can decode 7 objects well on the whole-brain EEG data using just the raw amplitude (same with our MEG-based analysis), and we can detect fast backward sequenceness (peaked at 30 ms time lag) during choice/planning time, similar with our previous MEG findings [53]. It is result from one subject, we are cautious to make serious claim, nevertheless we think it is promising.



Sequence detection in EEG data (from one participant). a, Task design. At each trial, the participant starts at sate A, and he/she will need to select either "BDF" or "CEG" path, based on the final reward receipt on state F and G. All seven states are indicated by pictures. b, The leave-one-out crossed validated decoding accuracy is shown, with peak around 200 ms after stimulus onset, similar with our previous MEG findings. c, TDLM method is applied on the decoded state time course and find a fast backward sequenceness following task structure. This is subtraction between forward and backward, therefore the negative sequenceness indicate stronger backward sequence. The dotted line is the peak of the absolute state-permutation at each time lag, the dotted line the max over all computed state time lags, for controlling multiple comparison. This is the same statistical method we have used in previous empirical work and this method paper. This sequence results in EEG replicates our previous MEG-based finding during planning/decision time (see Figure 3 in Kurth-Nelson et al., 2016, and also see Figure 3f in Liu et al., 2019).

APPENDIX 2: Pseudocode of cross-validations

In the consideration of the formatting, we have attached the Latex-based algorithm box in picture form.



Algorithm 1: hold one out cross validation to compute classification accuracy. Here N is number of trials, D is number of data dimensions, and P is number of classes:

```
Algorithm 1: Hold one out cross validation
input  : Data set $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N} (X_i \in \mathbb{R}^D; Y_i \in \mathbb{Z}_2^P)$
output: Cross validated classification accuracy $\{a \in \mathbb{R} : 0 \leq a \leq 1\}$
Randomly split $\mathcal{D}$ into $K = \frac{N}{P}$ equally sized subsets, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_K\}$ such
  that each $\mathcal{D}_i$ contains a single random sample from each class in $\mathcal{Y}$ ;
for $k$ in $K$ do
    Create a training dataset $\mathcal{T}_k = \{\mathcal{D}_i : i \neq k\}$ ;
    Train a logistic regression classifier $\beta_k$ on $\mathcal{T}_k$ ;
    Compute classification accuracy $a_k$ of $\beta_k$ on $\mathcal{D}_k$ ;
end
Compute mean accuracy $a = \frac{1}{K} \sum_{k=1}^{K} a_k$
```

Algorithm 2: test a classifier's abstraction ability across different datasets with some common structure.

```
Algorithm 2: Classifier Abstraction
input  : Data set $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N} (X_i \in \mathbb{R}^D; Y_i \in \{A, B, C, D, A', B', C', D'\})$
output: Abstraction accuracy $\{a \in \mathbb{R} : 0 \leq a \leq 1\}$
Partition $\mathcal{D}$ into two subsets each of which exclusively contain trials from one or
  other structure sequence: $\mathcal{D}_1 = \{X_i, Y_i\}_{i=1}^{N} (X_i \in \mathbb{R}^D; Y_i \in \{A, B, C, D\}$ and
  $\mathcal{D}_2 = \{X_i, Y_i\}_{i=1}^{N} (X_i \in \mathbb{R}^D; Y_i \in \{A', B', C', D'\}$ ;
for $k$ in { 1,2 } do
    Train a logistic regression classifier $\beta_k$ on $\mathcal{D}_k$ ;
    Compute classifier predictions $p_k$ of $\beta_k$ on $\mathcal{D}_{3-k}$ ;
    Compute abstraction accuracy $a_k$ as proportion of samples for which the
      prediction $p_k$ correctly identifies the sequence location (eg $A$ predicted for $A'$) ;
end
Compute mean abstraction accuracy $a = \frac{1}{2} \sum_{k=1}^{2} a_k$
```