# Temporally delayed linear modelling (TDLM) measures replay in both animals and humans

Yunzhe Liu,[1,2,3*] Raymond J Dolan,[1,3,4] Cameron Higgins,[5] Hector Penagos, [6] Mark Woolrich,[5] H. Freyja Ólafsdóttir,[7] Caswell Barry,[8] Zeb Kurth-Nelson,[3,9,10] Timothy Behrens [4,5,10]

1. State Key Laboratory of Cognitive Neuroscience and Learning, IDG/McGovern Institute for Brain Research, Beijing Normal University, Beijing, China
2. Chinese Institute for Brain Research, Beijing, China
3. Max Planck University College London Centre for Computational Psychiatry and Ageing Research, London, UK
4. Wellcome Centre for Human Neuroimaging, University College London, London, UK
5. Wellcome Centre for Integrative Neuroimaging, University of Oxford, Oxford, UK
6. Center for Brains, Minds and Machines, Picower Institute for Learning and Memory, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA
7. Donders Institute for Brain Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands
8. Research Department of Cell and Developmental Biology, University College London, London, UK
9. DeepMind, London, UK
10. Senior author

[*] Correspondence: yunzhe.liu@bnu.edu.cn (Y.Z.L.)

## SUMMARY

There are rich structures in off-task neural activity which are hypothesised to reflect fundamental computations across a broad spectrum of cognitive functions. Here, we develop an analysis toolkit – Temporal Delayed Linear Modelling (TDLM) for analysing such activity. TDLM is a domain-general method for finding neural sequences that respect a pre-specified transition graph. It combines nonlinear classification and linear temporal modelling to test for statistical regularities in sequences of task-related reactivations. TDLM is developed on the non-invasive neuroimaging data and is designed to take care of confounds and maximize sequence detection ability. Notably, as a linear framework, TDLM can be easily extended, without loss of generality, to capture rodent replay in electrophysiology, including in continuous spaces, as well as addressing second-order inference questions, e.g., its temporal and spatial varying pattern. We hope TDLM will advance a deeper understanding of neural computation and promote a richer convergence between animal and human neuroscience.

## INTRODUCTION

Human neuroscience has made remarkable progress in detailing the relationship between the representations of different stimuli during task performance [1-3]. At the same time, it is increasingly clear that resting, off-task, brain activities are structurally rich [4,5]. An ability to study spontaneous activity with respect to task-related representation is important for understanding cognitive process beyond current sensation [6]. However, unlike the case for task-based activity, little attention has been given to techniques that can measure representational content of resting brain activity in humans.

Unlike human neuroscience, representational content of resting activity is studied extensively in animal neuroscience. One seminal example is “hippocampal replay” [7-10]: During sleep, and quiet wakefulness, place cells in the hippocampus (that signal self-location during periods of activity) spontaneously recapitulate old, and explore new, trajectories through an environment. These internally generated sequences are hypothesized to reflect a fundamental feature of neural computation across tasks [11-15]. Numerous methods have been proposed to analyse hippocampal replay [16-18]. However, they are not domain general in that they are designed to be most suited for specific needs, such as particular task design, data modality, or research question [19,20]. Most commonly, these methods apply to invasive electrophysiology signals, aiming to detect sequences in a linear track during spatial navigation task [20]. As a result, they cannot be directly adapted for analysing human resting activity collected using non-invasive neuroimaging techniques. Furthermore, in rodent neuroscience, it is non-trivial to adapt these algorithms to even small changes in tasks (such as 2D foraging). This may be a limiting factor in taking replay analyses to more interesting and complex tasks, such as complex mazes [21].

Here, we introduce TDLM (temporal delayed linear modelling), a domain general analysis toolkit, for characterizing temporal structure of internally generated neural representations in rodent electrophysiology as well as human neuroimaging data. TDLM is inspired by existing replay detection methods [8,16,17], especially those analysis of population of relay events [17]. It is developed based on the General Linear Modelling (GLM) framework, and can therefore easily accommodate testing of "second-order" statistical questions [19], such as whether there is

more forward than reverse replay, or is replay strength changing over time, or differs between behavioural conditions. This type of question is ubiquitous in cognitive studies, but is typically addressed ad-hoc in other replay detection methods [19]. In TDLM, such questions are treated naturally as linear contrasts of effects in a GLM.

Here we show TDLM is suited to measure the **average** amount of replay across many events (i.e., replay strength) in linear modelling. This makes it applicable to both rodent electrophysiology and human neuroimaging. Applying TDLM on non-invasive neuroimaging data in humans, we, and others, have shown it is possible to measure the average sequenceness (propensity for replay) in spontaneous neural representations [22-25]. The results resemble key characteristics found in rodent hippocampal replay and inform key computational principles of human cognition [24].

In the following sections, we first introduce the logic and mechanics of TDLM in detail, followed by a careful treatment of its statistical inference procedure. We test TDLM in both simulation (see the Methods section "Simulating MEG data") and real human MEG/EEG data (see the Methods section "Human replay dataset"). We then turn to rodent electrophysiology and compare TDLM to existing rodent replay methods, extending TDLM to work on a continuous state space. Lastly, using our approach we re-analyse rodent electrophysiology data from Ólafsdóttir, et al. [26] (see the Methods section "Rodent replay dataset"), and show what TDLM can offer uniquely compared to existing methods in rodent replay analysis.

To summarise, TDLM is a general, and flexible, tool for measuring neural sequences. It facilitates cross-species investigations by linking large-scale measurements in humans to single neuron measurements in non-human species. It provides a powerful tool for revealing abstract cognitive processes that extend beyond sensory representation, potentially open doors for new avenues of research in cognitive science.

# RESULTS

## TDLM

### Overview of TDLM

Our primary goal is to test for temporal structure of neural representations in humans. However, to facilitate cross-species investigation [27], we also want to extend this method to enable measurement of sequences in other species (e.g., rodents). Consequently, this sequence detection method has to be domain general. We choose to measure sequences in a decoded state space (e.g., posterior estimated locations in rodents[17] or time course of task-related reactivations in humans[28]) as this makes results from different data types comparable.

Ideally, a general sequence detection method should (1) uncover structural regularities in the reactivation of neural activity, (2) control for confounds that are not of interest, and (3) test whether this regularity conforms to a hypothesized structure. To achieve these goals, we developed the method under a GLM framework, and henceforth refer to it as Temporal Delayed Linear Modelling, i.e., TDLM. Although TDLM works on a decoded state space, it

still needs to take account of confounds inherent in the data where the state space is decoded from. This is a 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 task variable of interest. This is what we call the 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-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) appear in a consistent order: $A \rightarrow B \rightarrow C \rightarrow D$ (Figure 1a, b). If we are interested in replay of this sequence during subsequent resting periods (Figure 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+$\Delta t$", 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 linear relation between the $X_i$ time series and the $\Delta t$-shifted $X_j$ time series. These $n^2$ relations comprise an empirical transition matrix, describing how likely each variable is to be succeeded at a lag of $\Delta t$ by each other variable (Figure 1e). Second, we linearly relate this empirical transition matrix with a task-related transition matrix of interest (Figure 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 $\Delta t$ of interest, yielding a measure of sequenceness at each possible lag between variables, and submit this for statistical inference (Figure 1g).

Note that, for now, this approach decomposes a sequence (such as $A \rightarrow B \rightarrow C \rightarrow D$) into its constituent transitions and sums the evidence for each transition. Therefore, it 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. For interested readers, we address how to strengthen the inference by looking explicitly for longer sequences in the Appendix 1: Multi-step sequences.

## Constructing the empirical transition matrix

In order to find evidence for state-to-state transitions at some time lag $\Delta t$, we could regress a time-lagged copy of one state, $X_j$, onto another, $X_i$ (omitting residual term ε in all linear equations):

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

(1)

Instead, TDLM chooses to include all states in the same regression model for important reasons, detailed in section "Moving to multiple linear regression":

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

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:

$$X(\Delta t) = X\beta$$

(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 $\Delta t$. $\beta_{ij}$ is an estimate of the influence of $X_i(t)$ on $X_j(t + \Delta t)$. $\beta$ is an $n \times n$ matrix of weights, which we call the *empirical transition matrix*.

To obtain $\beta$, 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 $\beta$ at every time lag. We call this step the first level sequence analysis.

**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, $T_F$. Therefore, we construct another GLM which relates $T_F$ to the empirical transition matrix, $\beta$. We call this step the second level sequence analysis:

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

As noted above, $\beta$ is the empirical transition matrix obtained from first-stage GLM. It has dimension of $n$ by $n$, where $n$ is the number of states. Each entry in $\beta$ reflects the unique contribution of state $i$ to state $j$ at given time lag. Effectively, the above equation models this empirical transition matrix $\beta$ as a weighted sum of *prespecified template matrices*, $T_r$. Thus, $r$ is the number of regressors included in the second-stage GLM, and each scalar valued $Z(r)$ is the weight assigned to the $r$ th template matrix. Put in other words, $T_r$ constitutes the regressors in the design matrix, each of which has a prespecified template structure, e.g., $T_{auto}$, $T_{const}$, $T_F$ and $T_B$ (Figure 1h).

$T_F$ and $T_B$ are the transpose of each other (e.g., red and blue entries in Figure 1b), indicating transitions of interest in forward and backward direction, respectively. In 1D 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 $T_B$ reflects its unique contribution. $T_{auto}$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 strength of the hypothesised 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 1f). $Z_F$, $Z_B$ are the forward and backward sequenceness respectively (e.g., red and blue lines in Figure 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 many times and re-calculating the second level analysis for each shuffle (Figure 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 incorrect ways to perform permutations, which permute factors that are not exchangeable under the null hypothesis and therefore lead to false positives. We 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 is discussed later.

If the time lag $\Delta t$ at which neural sequences exist is not known *a priori*, then we must 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 (see details in section "Correcting for multiple comparisons"). If we choose this test statistic, then any values of $Z_F$ exceeding the 95$^{th}$ percentile of the null distribution can be treated as significant at $\alpha = 0.05$ (e.g., the grey dotted line in Figure 1g).

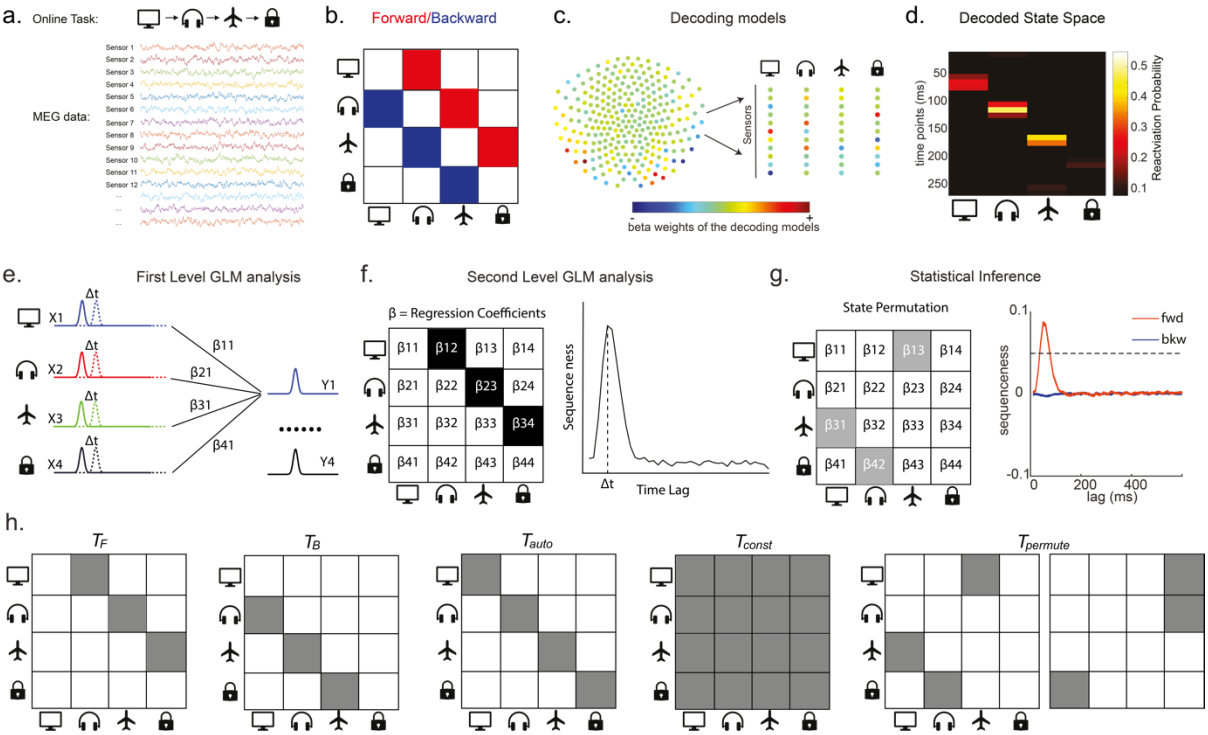



**Figure 1. Task design and illustration of TDLM. a,** Task design in both simulation and real MEG data. Assuming there is one sequence, A->B->C->D, indicated by the four objects at the top. During the task, participants are shown the objects, and asked to figure out a correct sequence for these objects while undergoing MEG scanning. A snapshot of MEG data is shown below. It is a matrix with dimensions of sensors by time. **b,** The transitions of interest are shown, with the red and blue entries indicating transitions in the forward and backward direction respectively. **c,** The first step of TDLM is to construct decoding models of states from 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 (empirical transition matrix), $\beta$ at each time lag. **f,** In the second-level GLM, this coefficient matrix is projected onto the hypothesized transition matrix (black entries), to give a single measure of sequenceness. Repeating this process for the number of time lags of interest generates sequenceness over time lags (right panel). **g,** The statistical significance of sequenceness 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 95$^{th}$ percentile of all shuffles on the maximum value over all tested time lags. **h,** The second level regressors $T_{auto}$, $T_{const}$, $T_F$ and $T_B$, as well as two examples of the permuted transitions of interest, $T_{permute}$ (for constructing permutation test), are shown.

## TDLM STEPS IN DETAIL

### Getting the states

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

### 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 associated neural representation, we present these stimuli in a randomized order at the start of a task, and record whole-brain neural activity using a non-invasive neuroimaging method, e.g., MEG or EEG. We then train a model to map the pattern of recorded neural activity to the presented image (Figure 1-figure supplement 1). This could be any of the multitude of available decoding models. For simplicity we used a logistic regression model throughout.

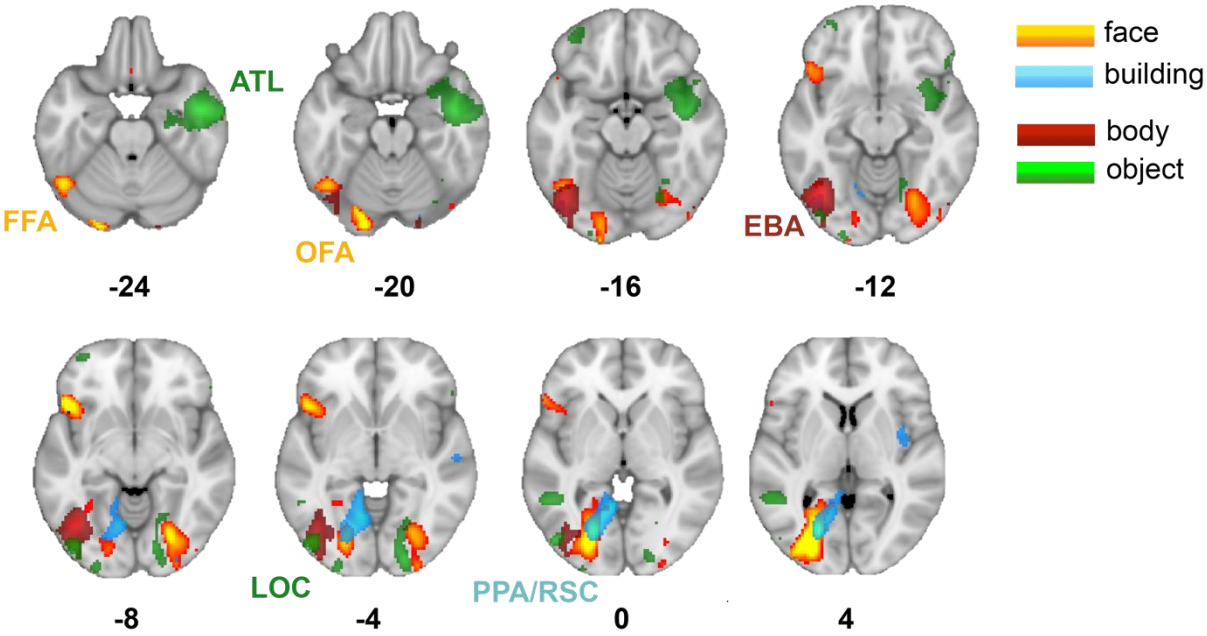



**Figure 1-figure supplement 1. Source localization of stimuli evoked neural activity in MEG.** The states here are defined in terms of stimuli evoked neural activity. The classifiers are trained at 200 ms post-stimulus onset. For example, the stimuli are faces, buildings, body parts, and objects. Source localizing the evoked neural activity, we found that the activation patterns of stimuli in MEG signal are consistent with those reported in fMRI 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 a parahippocampal place area (PPA) and retrosplenial cortex (RSC), a region also known to respond to scene and building stimuli. Activation for body part stimuli localised to a region consistent with the extrastriate body area (EBA). Activation for objects was in a region consistent with an object-associated lateral occipital cortex (LOC) as well as an anterior temporal lobe (ATL) cluster that may relate to conceptual processing of objects. Those maps are thresholded to display localized peaks. The full un-thresholded maps can be found at https://neurovault.org/collections/6088/. This is adapted from Wimmer, et al. [22].

In MEG/EEG, neural activity is recorded by multiple sensor 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 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 can be 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 features of stimuli, we can use a classical leave-one-out cross-validation scheme to determine the ability of classifiers to generalise to unseen data of the same stimulus type (decoding accuracy) at each time point (see Appendix 2 for its algorithm box). In essence, this cross-validation scheme is asking whether the classifier trained on this sensory feature can be used to classify the unseen data of the same stimuli (Figure 2a, b).

After we have identified the peak time point based on the cross validation, we can train the decoding models based on the multivariate sensor data at this given time.

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

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

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

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

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

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

**States as abstractions**

As well as sequences of sensory representations, it is possible to search for replay of more abstract neural representations. Such abstractions might be associated with the presented image (e.g., mammal vs fish), in which case analysis can proceed as above by swapping categories for images [22]. A more subtle example, however, is where the abstraction pertains to the sequence or graph itself. In space, for example, grid cells encode spatial coordinates in a fashion that abstracts over the sensory particularities of any one environment, and therefore can be reused across environments [29]. In human studies similar representations have been observed for the location in a sequence [24,30]. For example, different sequences have shared representations for their second items (Figure 2). These representations also replay [24]. However, to measure this replay we need to train decoders for these abstract representations. This poses a conundrum as it is not possible to elicit the abstract representations in the absence of the concrete sequence (the sensory stimuli). Care is required to ensure that the decoders are sensitive to the abstract code rather than the sensory representations (see Appendix 2 for algorithm box of selecting time point for training abstract code). Useful strategies include training classifiers to generalise across stimulus sets, and ensuring the classifiers are orthogonal to sensory representations (Figure 2-figure supplement 1 - details in Liu, et al. [24]). One way that excludes the possibility of sensory contamination is if the structural representations can be shown to sequence before the subjects have ever seen their sensory correlates [24].

TDLM can also be used iteratively to ask questions about the ordering of different types of replay events (Figure 2d). This can provide for powerful inferences about the temporal organisation 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 in the Appendix 3: Sequences of sequences.

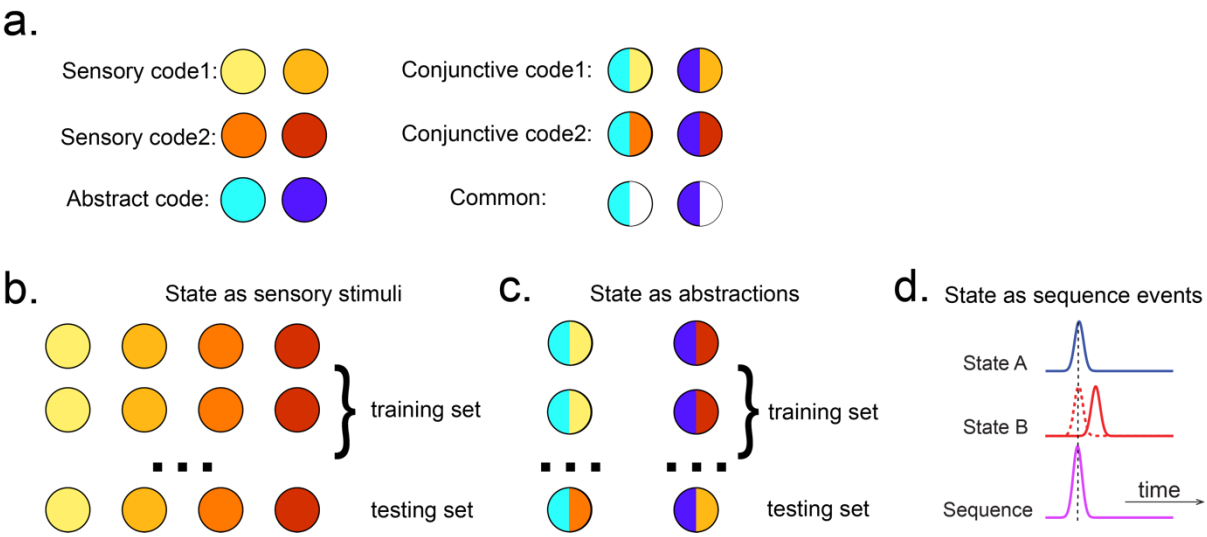



**Figure 2. Obtaining different state spaces. a**, Assuming we have two abstract codes, each abstract code has two different sensory codes (left panel). The M/EEG data corresponding to each stimulus is a conjunctive representation of sensory and abstract codes (right panel). The abstract code can be operationalised as the common information in the conjunctive codes of two stimuli. **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 abstracted information. The state can also be defined as the abstractions. To extract this information, we need to avoid a confound of sensory information. We can train the classifier on the neural activity evoked by one stimulus and test it on the other sharing 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**, The state can also be defined as the sequence event itself.

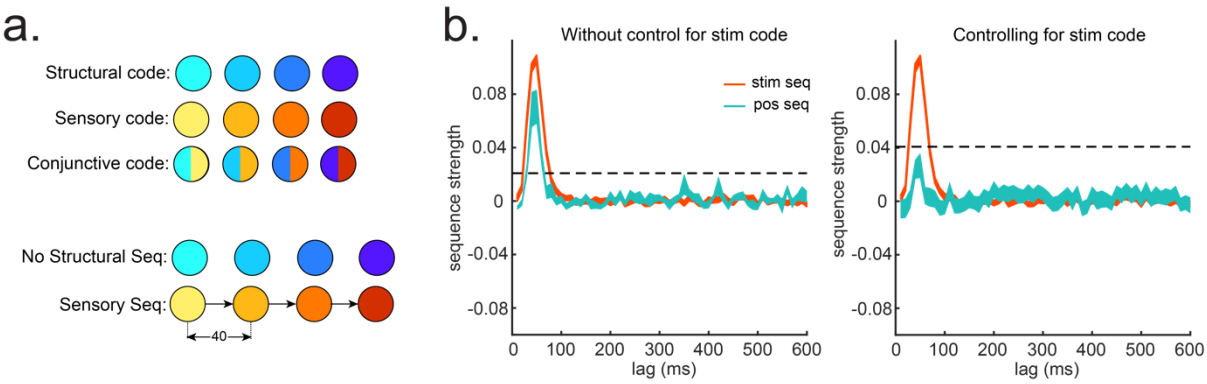



**Figure 2-figure supplement 1. Sequences of abstract code. a**, Illustration of the relationship between sensory code and (abstract) structural code. Structural code cannot be accessed directly but

can be indirectly obtained from the conjunctive code (overlapping representation of sensory and structural code). In this simulation, there is sequence of sensory code but not of structural code. **b,** We show the importance of controlling for sensory (stim) information when looking for sequences of abstract code: If sensory information is not controlled, we would observe significant sequences of structural code, while in fact, it is not present, i.e., false positive.

## Controlling confounds and maximising sensitivity in sequence detection

Here, we motivate the key features of TDLM.

### Temporal correlations

In standard linear methods, unmodelled temporal autocorrelation can inflate statistical scores. Techniques such as auto-regressive noise modelling are commonplace to mitigate these effects [31,32]. However, autocorrelation is a particular burden for analysis of sequences, where it interacts with correlations between the decoded neural variables.

To see 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 autocorrelations, 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 spuriously report 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 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 the difference measure described above ($D = Z_F - Z_B$). This works well as shown in Kurth-Nelson, et al. [11]. However, a downside is it prevents us from measuring forward and backward sequences independently. The remainder of this section considers alternative approaches that allow for 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 can solve the problem by simply controlling for it in a linear regression, as in Granger Causality [33]:

$$X_j(t+\Delta t) = \beta_0 + X_i(t)\beta_{ij} + X_j(t)\beta_{jj} \tag{9}$$

Unfortunately, we do not have access to the ground truth of $X$ because 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 is imperfect, leading to spurious weight on $\beta_{ij}$, 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 control for that whole subspace, we would be on safe ground. 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 3a 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 too in cross correlation. This is why previous studies based on a cross-correlation [34,35] cannot look for sequenceness in forward and backward directions separately, but have to rely on their asymmetry. The multiple regression accounts for the correlation structure of the data and allows correct inference to be made. Unlike the simple subtraction method proposed above (Figure 3a, left panel), the multiple regression permits separate inference on forwards and backwards sequences.

**Oscillations and long timescale autocorrelations:** 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 a large neural oscillation. For example, during rest (which is often the time 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 oscillations at different phases. This scenario may occur, for example, if we assume there are travelling waves in cortex [36,37], because different sensors will experience the wave at different times, and different states have different contributions from each sensor. MEG sensors can be seen as measures of local field potential on the scalp, which contain background neural oscillations. In humans this is dominantly alpha during rest.

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 $\Delta t$ is $\frac{1}{4}\,\tau$ (where $\tau$ 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, we can include phase shifted versions/more timepoints of $X(t)$. If dominant background oscillation is at alpha frequency (e.g., 10Hz), neural activity at time T would be correlated with activity at time T + $\tau$. We can control for that, by including $X(t+\tau)$, as well as $X(t)$ in the GLM (Figure 3b). Here $\tau$ = 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 [24].

**Spatial correlations**

As mentioned above, correlations between decoded variables commonly occur. 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 correlations between states reduces the sensitivity of sequence detection, because it is difficult to distinguish between states within the sequence: collinearity impairs estimation of $\beta$ in Equation 2. In Figure 3c, we show in simulation that the ability to detect real sequences goes down as the absolute value of a 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 null data [24,35]. This 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 3d, we show in simulation the ensuing benefit for sequence detection. An alternative method is penalizing covariance between states in the classifier's cost function [38].

**Regularization**

A key parameter in training high dimensional decoding models is the degree of regularization. In sequence analysis, we are often interested in spontaneous reactivation 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 well suited for decoding spontaneous activity at subsequent tests. Regularising the classifier (for example with an L1 Norm) is a common technique for increasing out-of-sample generalisation (to avoid overfitting). Here it has the added potential benefit of reducing spatial correlation between classifier weights.

During classifier training, we can impose L1 or 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 regularization term. L1 regularization can be phrased as maximising the likelihood, subject to a regularisation penalty on the L1 norm of the coefficient vector:

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

(10)

L2 regression can be viewed as a problem of maximising the likelihood, subject to a regularisation penalty on the L2 norm of the coefficient vector:

$$W = \mathrm{argmax}_W \left[ \log\left(P(Y|M,W)\right) + b\,\lambda_{L2}\,||\,W\,||_2 \right] \quad (11)$$

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. $P(Y|M,W) = \sigma(MW)$, and $\sigma$ is the logistic sigmoid function.

We simulate data with varying numbers of true sequences at 40 ms lag, and we find the beta estimate of sequence strength at 40 ms positively relates to the number of sequences. We also find that L1 weight regularization is able to detect sequences more robustly than L2 regularization, while L2 performs no better than an unregularized model (Figure 3e). The L1 models also have much lower spatial correlation, consistent with L1 achieving better sequence detection by reducing the covariances between classifiers [39].

In addition to minimizing spatial correlations, as discussed above, it can also be shown that L1-induced sparsity encodes weaker assumptions about background noise distributions into the classifiers, as compared to L2 regularization [39]. This might be of special interest to researchers who want to measure replay during sleep. Here, the use of sparse classifiers is helpful as background noise distributions are likely to differ more substantially from the (awake state) training data.

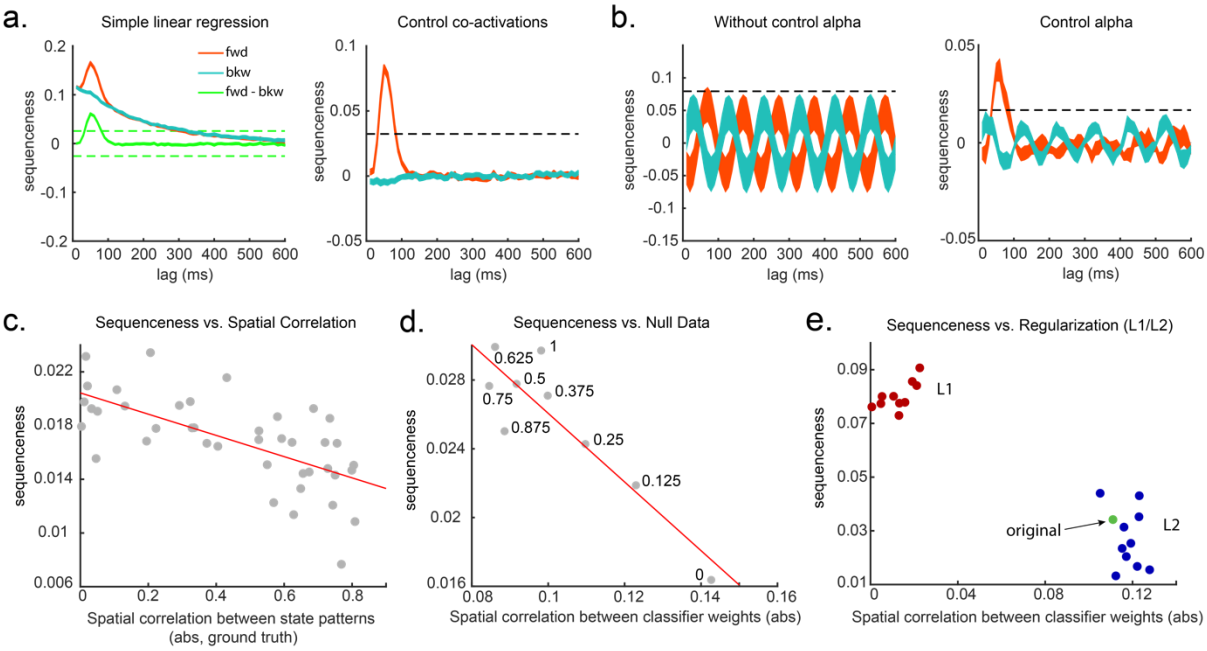



**Figure 3. Effects of temporal, spatial correlations, and classifier regularization on TDLM. a,** Simple linear regression or cross-correlation approach relies on an asymmetry of forward and backward transitions; therefore, subtraction is necessary (left panel). TDLM instead relies on multiple linear 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 the sensitivity of sequence detection. This suggests reducing overlapping patterns between states is important for sequence detection. **d,** Adding null data to the training set increases 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).

## STATISTICAL INFERENCE

So far, we have shown how to quantify sequences in representational dynamics. An essential final step is 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 follows repeating dynamical motifs [40]. 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 a state-to-state lag of interest ($\Delta t$) is not known a priori, it is then 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.

### Distribution of sequenceness at a single lag

If a state-to-state lag of interest ($\Delta t$) 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 the data are centred on zero if there were no real sequences. We show this approach is safe in both simulation (assuming no real sequences) and 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 compare false positive rates predicted by the statistical tests with false positive rates measured in simulation (Figure 4a). We see the empirical false positives are well predicted by theory.

We have tested this also on real MEG data. In Liu, et al. [24] we had one condition where we measured resting activity before the subjects saw any stimuli. Therefore, by definition these sensory stimuli could not be replayed, we can use classifiers from these stimuli (measured later) to test a false positive performance of statistical tests on replay. Note, in our case, each

subject saw the same stimuli in a different order. They could not know the correct stimulus order when these resting data were acquired. These data provide a valid null for testing false positives.

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 a signed rank test or one-sample *t* test across subjects. Again, predicted and measured false positive rates match well (Figure 4b, left panel). This holds true across all computed time lags (Figure 4b, right panel).

An alternative to making assumptions about the form of the null distribution is to compute an empirical null distribution by permutation. Given that we are interested in the sequence of states over time, one could imagine permuting either state identity or time. However, permuting time uniformly will typically lead to a very high incidence of false positives, as time is not exchangeable under the null hypothesis (Figure 4c, blue colour). Permuting time destroys the temporal smoothness of neural data, creating an artificially narrow null distribution [24,35]. This false positive also exists if we circular shift the time dimension of each state, rather than randomly permuting the state identities. This is because the signal is highly non-stationary. Replays come in bursts, as recently analysed [6], and this will break a circular shift [41]. 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 4c, purple colour).

**Correcting for multiple comparisons**

If the state-to-state lag of interest is not known, we have to search over a range of time lags. As a result, we then have a multiple comparison problem. Unfortunately, we don't as yet have a good parametric method to control for multiple testing over a distribution. It is possible that one could use methods that exploit the properties of Gaussian Random Fields, as is common in fMRI [42], but we have not evaluated this approach. Alternatively, we could use Bonferroni correction, but the assumption that each computed time lag is independent is likely false and overly conservative.

We recommend relying on state-identity based permutation. To control for the family wise error rate (assuming $\alpha = 0.05$), we want to ensure 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 approach as used for family-wise error correction for permutations tests in fMRI data [43], and in our case it is shown to behave well statistically (Figure 4d).

**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 implemented in Kurth-Nelson, et al. [35]. 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 implemented in Liu, et al. [24]. In each case, permutations are equivalent to the test data under the assumption that states are exchangeable between positions and sequences. The first case 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 a signal, as the null distribution is guaranteed not to include permutations which share any transitions with the test data (Figure 4e). For example, in the Figure 4e, the blue swaps are the permutations that only exchange state identity across sequences, as in Liu, et al. [24]; while the red swaps are the permutations that permit all possible state identity permutations, as in Kurth-Nelson, et al. [35]. Note there are 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.

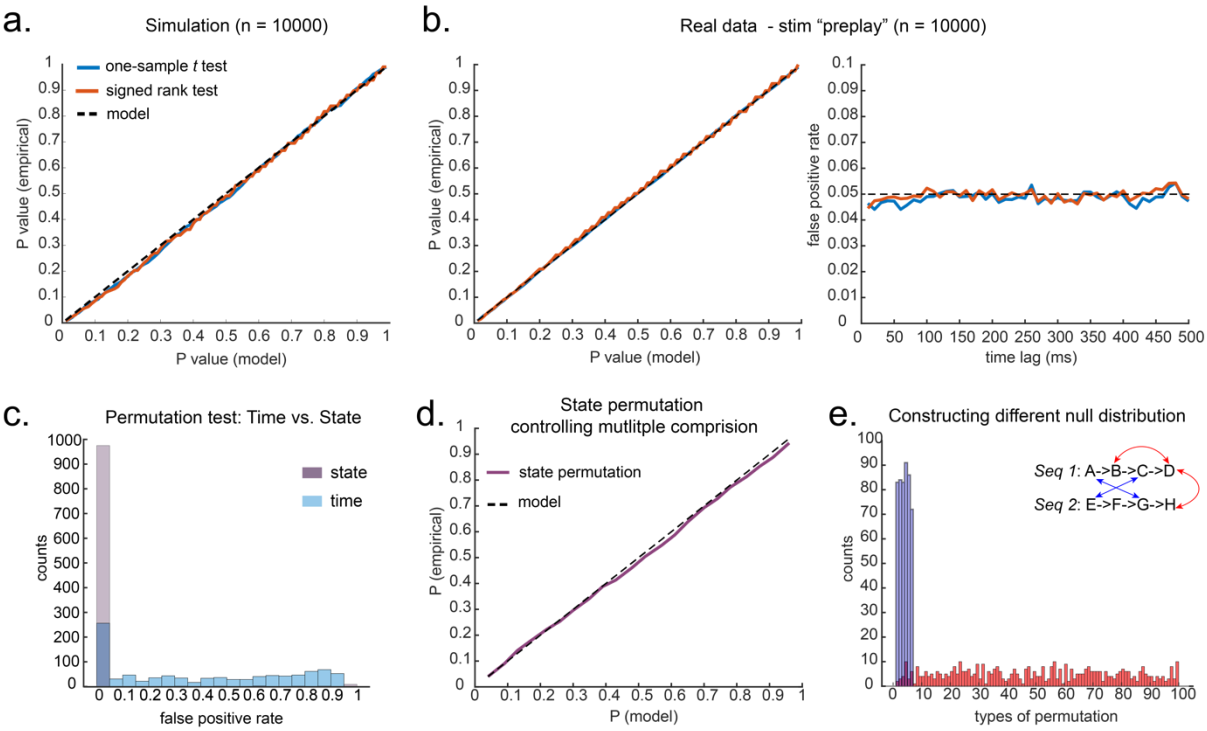



**Figure 4. Statistical inference. a,** P-P plot of one-sample *t* test (blue) and Wilcoxon signed rank test (red) against zero. This is performed in simulated MEG data, assuming auto-correlated state time courses, but no real sequences. In each simulation, the statistics are done only on sequenceness at 40 ms time lag, across 24 simulated subjects. There are 10,000 simulations. **b,** We have also tested the sequenceness distribution on real MEG data. Illustrated is the pre-task resting state on 22 subjects from Liu et. al, where the ground truth is the absence of sequences given the stimuli have not yet been shown. The statistics are done on sequenceness at 40 ms time lag, across the 22 subjects. There are eight states. The state identity is randomly shuffled 10,000 times to construct a null distribution. **c,** Time-based permutation test tends to result in 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. There are 10,000 simulations. In each simulation, there are 24 simulated subjects, with no real sequence. **e,** In state-identity based permutation, we can test more specific

hypotheses by controlling the null distribution. Blue are the permutations that only exchange state identity across sequences. Red are the permutations that permit all possible state identity permutations. 500 random state permutations are chosen from all possible ones. The X axis is the different combinations of the state permutation. It is sorted so that the cross-sequence permutations are in the beginning.

## Cautionary note on exchangeability of states after training

Until now, all non-parametric 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, take 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 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 may 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 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, behavioural pairing of stimuli can indeed result in increased neuronal similarity [44,45]. These early papers involved long training in monkeys. More recent studies have shown induced representational overlap in human imaging within a single day [28,46,47]. However, when analysed across the whole brain, such representational changes tend to be localised to discrete brain regions [48,49], and as a consequence may 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.

## SOURCE LOCALIZATION

Uncovering temporal structure of neural representation is important, but it is of interest to ask where in the brain a sequence is generated. Rodent electrophysiology research focuses mainly on the 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 in its ability to examine neural activity in

multiple other brain regions. Ideally, we would like a method that is capable of localizing sequences of more abstract representation in brain regions beyond hippocampus [24].

We want to identify the *time* when a given sequence is very likely to unfold, so we can construct averages of independent data over these times. We achieve this, by transforming from the space of original states, $X_{orig}$, to the space of sequence events, $X_{seq}$. First, based on the transition of interest, $T$, we can obtain the projection matrix, $X_{proj}$:

$$X_{proj} = X_{orig} \times T \tag{12}$$

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

$$X_{lag} = X_{orig}(t - \Delta \mathrm{t}) \tag{13}$$

Then, we obtain state space with sequence event as states by elementwise multiply $X_{proj}$ and $X_{lag}$ :

$$X_{seq} = X_{lag} .* X_{proj} \tag{14}$$

Each element in $X_{seq}$ indicates the strength of a (pairwise) sequence at a given moment in time. At this stage, $X_{seq}$ is a matrix with number of time points as rows (same as $X_{orig}$), and with number of pairwise sequences (e.g., A->B; B->C; etc) as columns. Now on this matrix, $X_{seq}$, we can either look for sequences of sequences (see in Appendix 3), or sum over columns (i.e., average over pairwise sequence events), and obtain a score at each timepoint reflecting how likely it is to be a sequence member (Figure 5a).

We can use this score to construct averages of other variables that might co-vary with replay. For example, if we choose timepoints when this score is high (e.g., 95$^{th}$) percentile after being low for the previous 100 ms and construct an average time-frequency plot of the raw MEG data aligned to these times, we can reconstruct a time-frequency plot that is, *on average*, associated with replay onset (Figure 5b). Note that although this method assigns a score for individual replay events as an intermediary variable, it results in an *average* measure across many events.

This approach is similar to spike-triggered averaging [50,51]. Applying this to real MEG data during rest, we can detect increased hippocampal power at 120-150 Hz, at replay onset (Figure 5b, c). Source reconstruction in the current analysis was performed using linearly constrained minimum variance (LCMV) beamforming, a common method for MEG source localization, but it is known to suffer from distal correlated sources [52]. A better method may be Empirical Bayesian Beamfomer for accommodating correlated neural source as a priori [53].

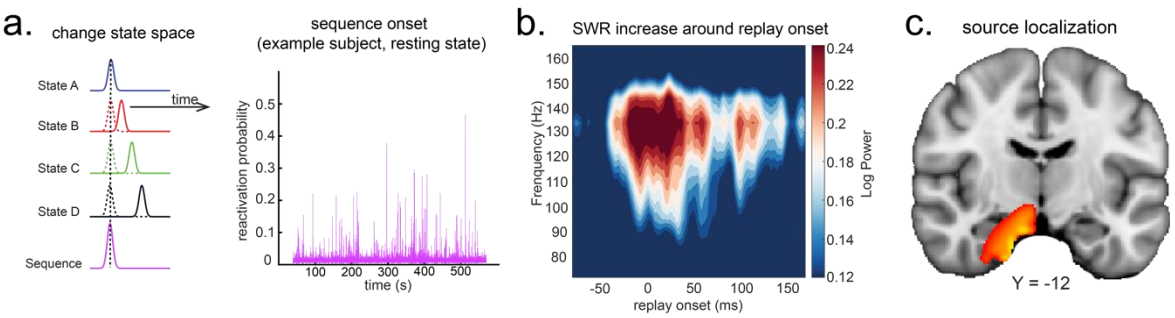



**Figure 5. Source localization of replay onset. a,** TDLM indexes the onset of a sequence based on the identified optimal state-to-state time lag (left panel). Sequence onset during resting state from one example subject is shown (right panel). **b,** There was a significant power increase (averaged 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 is reproduced from Figure 7 A, C, Liu et al. 2019, Cell, published under the Creative Commons Attribution 4.0 International Public License (CC BY 4.0).

## TDLM FOR RODENT REPLAY

So far, we have introduced TDLM in the context of analysing human MEG data. Relatedly, its application on human EEG data was also explored (Appendix 4: Apply TDLM to human whole-brain EEG data). Historically, replay-like phenomena have been predominantly studied in rodents with electrophysiology recordings in the hippocampal formation [16,17,20]. This raises an interesting question: how does TDLM compare to the existing rodent replay methods; can TDLM be applied to spiking data for detecting rodent replays, and what are the pros and cons? In this section we address this question.

### Generality of graph- vs line-based replay methods

Given TDLM works on the decoded state space, rather than sensor (with analogy to cell) level, we compared TDLM to rodent methods that work on the posterior decoded position (i.e., state) space, normally referred to as Bayesian-based methods [20] (Note that these methods are typically Bayesian in how position is decoded from spikes [54] but not in how replay is measured from decoded position). Two commonly used methods are Radon transform [16] and linear weighted correlation [17].

Both methods proceed by forming a 2D matrix, where one dimension is the decoded state (e.g., positions on a linear track), and the other dimension is time (note that the decoded state is embedded in 1D). The methods then try to discover if an ordered line is a good description of the relationship between state and (parametric) time. For this reason, we call this family of approaches “line search”.

The radon method uses a discrete Radon transform to find the best line in the 2D matrix [55] and then evaluates the radon integral, which will be high if the data lie on a line (Figure 6a).

It compares this to permutations of the same data where the states are reordered [20]. The linear weighted correlation method computes the average correlation between the time and estimated position in the 1D embedding (Figure 6b). The correlation is non-zero provided there is an orderly reactivation along the state dimension.

Both methods are applied to decoded positions, where they are sorted based on the order in a linearized state space. TDLM also works on the decoded position space, but instead of directly measuring the relationship between position and time, it measures the transition strength for each possible state to state transitions (Figure 6c).

This is a key difference between TDLM and these popular existing techniques. To reiterate, the latter rely on a continuous parametric embedding of behavioural states and the relationship between this embedding and time (parametrically encoded). TDLM is fundamentally different as it works on a graph and examines the statistical likelihood of some transitions happening more than others. This is therefore a more general approach that can be used for sequences drawn from any graph (e.g., 2D maze, Figure 6d), not just graphs with simple embeddings (like a linear track). For example, in a non-spatial decision-making task [35], all states lead to two different states and themselves can be arrived at from two other different states (Figure 6e). Existing "line search" methods will not work because there is no linear relationship between time and states (Figure 6f).

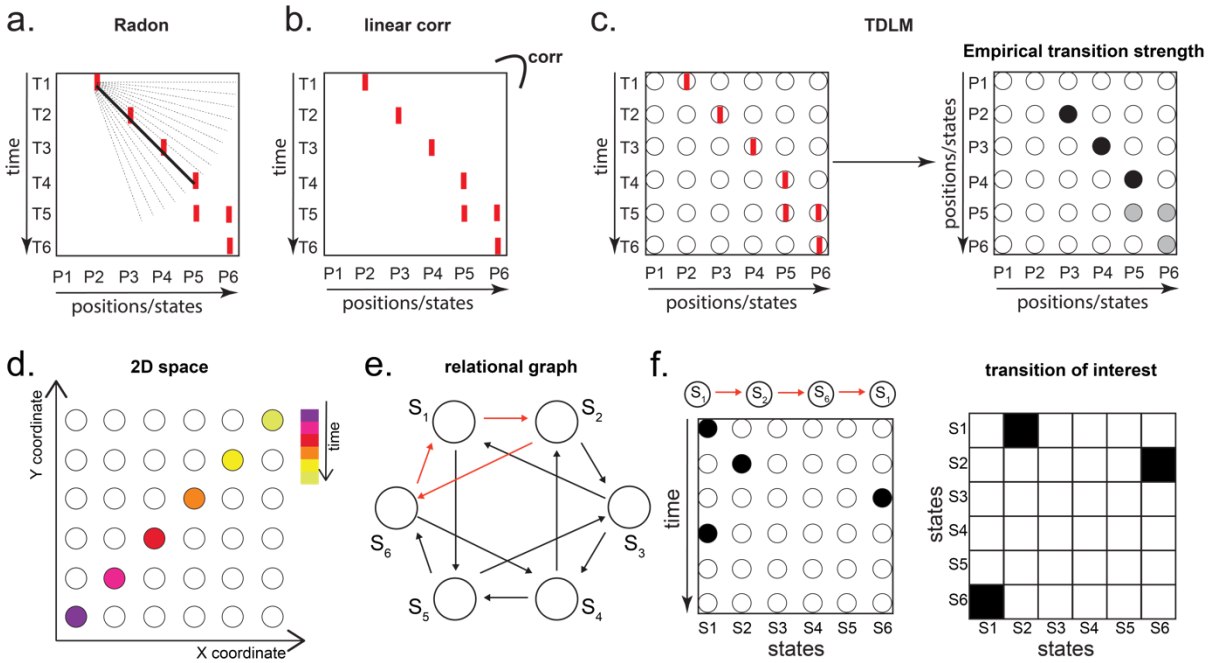



**Figure 6. TDLM vs. existing rodent replay methods. a.** The Radon method tries to find the best fitting line (solid line) of the decoded positions as a function of time. The red bars indicate strong reactivation at a given location. **b.** The linear correlation method looks for correlations between time and decoded position. **c.** The TDLM method, on the other hand, does not directly measure the relationship between state and time, but quantifies the likelihood of each transition. In the right panel, likelihood is indicated by darkness of shading. For example, P5 can be followed by either P5 or P6, making each transition half as likely as the deterministic P4->P5 transition. Later this empirical transition matrix is compared to a theoretical one, to quantify the extent to which the empirical transitions fit with a hypothesis. **d.** Sequences in 2D space is in three dimensions, which is hard to translate into a line search problem, e.g., time*position spaces. **e.** This is the transition matrix used in Kurth-Nelson, et al. [35], which cannot be translated into a linear state space. The transitions in red are

an example of a trajectory. **f**. Putting the example trajectory into the time by state matrix, we can see there is no linear relationship between them (left panel). In TDLM, this is tested by forming a hypothesis regressor in the state-to-state transition matrix (right panel).

## Multi-scale TDLM

While continuous spaces can be analysed in TDLM by simply chunking the space into discrete states, TDLM in its original form may potentially be less sensitive for such analyses than techniques with build-in assumptions about the spatial layout of the state space, such as the linear relationship between time and reactivated states (Appendix 5 "Less sensitivity of TDLM to skipping sequences"). In essence, because TDLM works on a graph, it has no information about the Euclidean nature of the state space, while techniques that make assumptions about the linear relationship between space and time benefit from these assumptions. For example, detecting state 1 then state 5 then state 10 counts as replay in these techniques, but not in TDLM.

However, TDLM can be extended to address this problem. For continuous state spaces, we first need to decide how to best discretise the space. If we choose a large scale, we will miss replays that occur predominantly within a spatial bin. If we choose a small scale, we will miss transitions that jump spatial bins. A simple solution is to apply TDLM at multiple different scales and take a (variance-weighted) average of the sequenceness measures at different scales. For example, when measuring replay speed, we can average events that travel 5 cm in 10 ms together with events that travel 10 cm in 20 ms.

Specifically, to perform multi-scale TDLM, we discretise position bins at multiple widths. This generates rate maps at multiple scales (e.g., 5 cm, 10 cm, 20 cm, 40 cm), and hence a multi-scale state space. For each replay speed of interest, we apply TDLM separately at each scale, and then take a variance-weighted average of replay estimates over all scales.

$$\beta_M = \frac{\sum_{i=1}^{n} \boldsymbol{\beta_i / V_i}}{\sum_{i=1}^{n} \mathbf{1} / \boldsymbol{V_i}}$$

(15)

Where $\beta_i$ is the sequence strength of given speed (i.e., state-to-state lag) measured at scale $i$, $V_i$ is the variance of its $\beta_i$ estimator, and $n$ is the number of scales. In the end, statistical testing is performed on the precision weighted averaged sequence strength, $\beta_M$, in the same way as we do in the original TDLM.

It is easy to see why this addresses the potential concerns raised above as some scales will capture the 1->2->3 transitions, whilst others will capture the 1->10->20 transitions: Because the underlying space is continuous, we can average results of the same replay speed together, and this will reinstate the Euclidean assumptions.

## Applying multi-scale TDLM to real rodent data (place cells in CA1)

We demonstrate the applicability of multi-scale TDLM by analyzing CA1 place cell spiking data from Ólafsdóttir, et al. [26]. In Ólafsdóttir, et al. [26], rats ran multiple laps on a 600 cm Z maze, and were then placed in a rest enclosure for 1.5 hours (Figure 7a). The Z maze consists

of 3 tracks, with its ends and corners baited with sweetened rice to encourage running from one end to the other. The animal's running trajectory was linearized, dwell time and spikes were binned into 2 cm bins and smoothed with a Gaussian kernel ($\sigma = 5$ bins). We generated rate maps separately for inbound (track1->track2->track3) and outbound (track3->track2->track1) running (see details in section "Rodent Replay dataset" in the Methods).

As in Ólafsdóttir, et al. [26], cells recorded in CA1 were classified as place cells if their peak firing field during track running was above 1 Hz with a width of at least 20 cm (see an example in Figure 7b). The candidate replay events were identified based on multi-unit (MU) activity from place cells during rest time. Periods exceeding the mean rate by 3 standard deviations of MU activity were identified as possible replay events. Events less than 40 ms long, or which included activity from less than 15% of the recorded place cell ensemble, were rejected (see an example of putative replay event in Figure 7c), and the remaining events were labelled putative replay events.

We analyzed data from one full recording session (track running for generating rate map, post-running resting for replay detection) from Rat 2192 reported in Ólafsdóttir, et al. [26]. Following the procedure described above, we identified 58 place cells, and 1183 putative replay events. Replay analysis was then performed on the putative replay events, separately for inbound and outbound rate maps given the same position has a different decoded state depending on whether it was during an outbound or inbound run.

A forward sequence is characterised by states from the outbound map occurring in the outbound order, or states from the inbound map occurring in the inbound order. Conversely, a backward sequence is when states from the inbound map occur in the outbound order or states from the outbound map occur in the inbound order. Candidate events were decoded based on a rate map, transforming the ncells * ntime to nstates * ntime. Each entry in this state space represents the posterior probability of being in this position at a given time. Replay analysis was performed solely on this decoded state space.

Note, TDLM is applied directly to the concatenated rather than individual replay events. This is because TDLM is a linear modelling framework. Applying TDLM on each single replay event, and then averaging the beta estimates (appropriately weighted by the variances) is equivalent to running TDLM once on the concatenated replay events. It quantifies the average amount of replay across many events, this is different compared to existing replay methods that focus on single replay events. Because TDLM addresses statistical questions in linear modelling, it does not require secondary statistics to ask whether the "counts" of individual events are more likely than chance, or more likely in one situation than another.

During the whole sleep period, TDLM identified a significant forward sequence for the outbound map with a wide speed range around from 1 to 10 m/s (Figure 7d, left panel), consistent with recent findings from Denovellis, et al. [56] on varying replay speed (similar results were obtained for inbound map, not shown here for simplicity). In our analysis, the fastest speed is up to 10 m/s, which is around 20X faster than its free running speed, representing approximately half a track-arm in a typical replay event, consistent with previous work [10,16,57,58].

**Second order inferences**

As pointed out by van der Meer, et al. [19], there are two types of statistical questions: a "first-order" sequence question, which concerns whether an observed sequenceness is different from random (i.e., do replays exist?); and a “second-order” question, which requires a comparison of sequenceness across conditions (i.e., do replays differ?). Because it is embedded in a linear regression framework, TDLM is ideally placed to address such questions. There are two ways of asking such questions in linear modelling - **Contrasts** and **Interactions**. We explain them with examples here.

**Linear contrasts**

After fitting a regression model, resulting in coefficients for different regressors, we can test hypotheses about these coefficients by constructing linear combinations of the coefficients that would be zero under the null hypothesis. For example, if we want to test whether effect **A** is greater than effect **B** then we can compute the linear contrast **A - B** (which would be zero under the null hypothesis) and perform statistics on this new measure. If we want to test whether replay increases linearly over 5 conditions [A, B, C, D, E], we can compute the linear contrast -2*A - B + 0*C + D + 2*E, (which would be zero under the null hypothesis) and perform statistics on this new measure. Statistics (within or across animals) can operate with these contrasts in exactly the same way as with the original coefficients from the linear model. Here we demonstrate this by showing in our example data set that there was a greater preponderance for forward than backward replay. We construct the contrast (Forwards - Backwards) and test it against zero using a multiple-comparison-controlled permutation test (Figure 7d, right panel, pink line). By constructing a different contrast (Forwards + Backwards), we can also show that the total replay strength across both types of replays was significant (Figure 7d, right panel, green line).

**Interactions**

A second method for performing second order tests is to introduce them into the linear regression as interaction terms, and then perform inference on the regression weights for these interactions. This means changing equation 2 to include new regressors. For example, if interested in how reactivations change over time, one could build new regressors ($Xtime_k(t)$), obtained by element-wise multiplying the state regressor, e.g., $X_k(t)$ with time indices ($Xtime_k(t) = X_k(t).*$ time). Now the first level GLM is constructed as (omitting residual term ε, same as equation 2):

$$X_j(t + \Delta t) = \sum_{k=1}^{n} X_k(t)\beta_{kj} + Xtime_k(t)\beta t_{kj} \qquad (16)$$

Example regressors in the design matrix can be seen in Figure 7e below. The first regressor, $X_k(t)$, is one of the state reactivation regressors used in standard TDLM. The second regressor, $Xtime_k(t)$, is this same as $X_k(t)$ multiplied by time. (There are k regressors of each form in regressor matrix.) Here, we chose to demean the time regressor before the interaction, so the early half of the regressor is negative, and the late half is positive. This has no effect on the regression coefficients of the interaction term but, by rendering the interaction approximately orthogonal to $X_k(t)$, it makes it possible to estimate the main effect and the interaction in the same regression.

Note that the interaction regressor is **orthogonal** to the state reactivation regressor, so it will have no effect on the first order regression terms. If we include such regressors for all states, then we can get two measures for each replay direction (sequence effect and time effect). The first tells us the average amount of replay throughout the sleep period (first order). The

second tells us whether replay increases or decreases as time progresses through the sleep period (second order).

**Orthogonal tests in regions of interest**

When examining Forward-Backward replay above, we did separate inference for each replay speed, and then performed multiple comparison testing using the max-permutation method (see statistical inference section above). We now take the opportunity to introduce another method common in human literature.

To avoid such multiple comparison correction, it is possible to select a "Region of Interest" (ROI), average the measure in question over that ROI, and perform inference on this average measure. Because we are now only testing one measure, there is no multiple comparison problem. Critical in this endeavour, however, is that we do not use the measure under test, or anything that correlates with that measure as a means to define the ROI. This will induce a selection bias [59]. In the example in Figure 7f, we have used the average replay (Forwards +Backwards) to select the ROI. We are interested in speeds in which there is detectable replay on average across both directions and the whole sleep period (Figure 7d, right panel, green shaded area). If we select our ROI in this way, we cannot perform unbiased inference on first order Forwards or Backwards replay because Forwards and Backwards regressors correlate with their sum (Figure 7d, statistical inference in the red rectangle is biased). However, we **can** perform unbiased inference on several second order effects (Figure 7d, statistical inference in the green rectangle). We can test (Forwards - Backwards) assuming the difference of terms is orthogonal to their sum (as it is in this case). Further we can test any interaction with time, because the ROI is defined on the average over time, and the interaction looks for ***differences*** as a function of time. When we perform these tests in our example dataset (Figure 7d, green rectangle), we confirm that there are more forward than backward replay on average. We further show that forward replay is decreasing with time during sleep, and that backward replay is increasing with time. Their difference (Forwards - Backwards) is also significant.

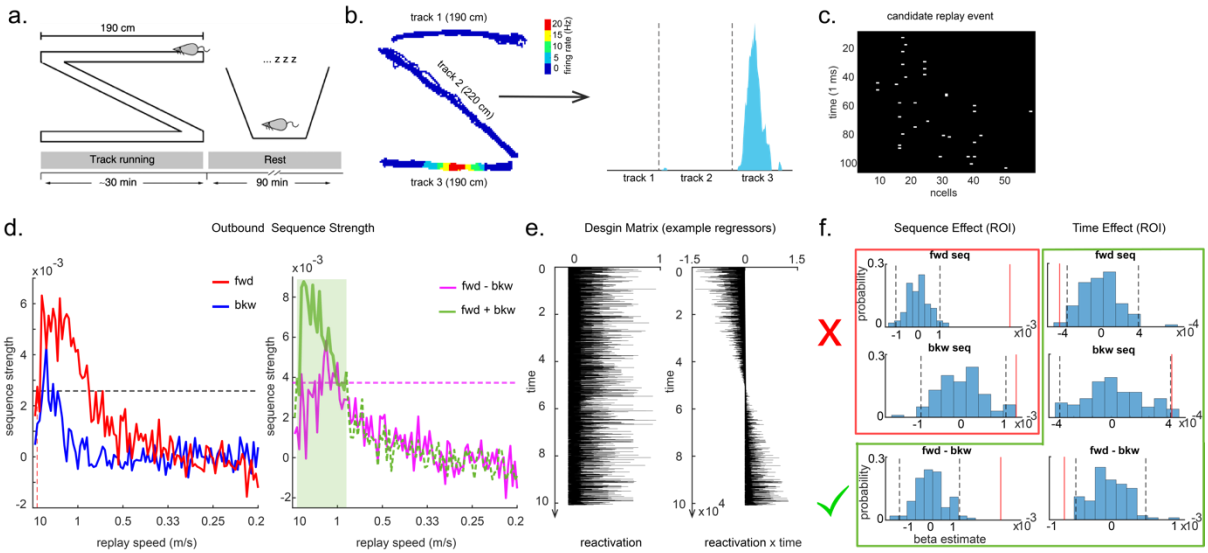



**Figure 7. TDLM applied to real rodent data. a**, The experimental design of Ólafsdóttir, et al. [26]. Rats ran on Z maze for 30 mins, followed by 90 min rest. **b,** An example rate map for a place cell is shown. The left panel shows its spatial distribution on the Z maze, and the right panel is its linearized distribution. **c,** An example of a candidate replay event (spiking data) is shown. **d,** Sequence strength

as a function of replay speed is shown for the outbound rate map. Black dotted line is the permutation threshold after controlling for multiple comparisons. Left panel: forward sequence (red) and backward sequence (blue). The red dotted line indicates the fastest replay speed that is significant – 10 m/s. Right panel: forward – backward sequence. The pink dotted line indicates the multiple comparison corrected permutation threshold for the replay difference. The green line is the sum of sequence strength between forward and backward direction. The solid line (with green shading) indicates the significant replay speeds (0.88 – 10 m/s) after controlling for multiple comparisons. We use this as a ROI to test for time varying effect on replay in panel f. **e,** Illustration of two exemplar regressors in the design matrix for assessing time effect on replay strength. The "reactivation" regressor is a lagged copy of reactivation strength of given position and is used to obtain sequence effect. The "reactivation x time" regressor is the elementwise multiplication between this position reactivation and time (z-scored), it explicitly models the effect of time on sequence strength. Both regressors are demeaned. **f,** Beta estimate of the sequence effect (left panel), as well as time modulation effect on sequence (right panel) in the ROI are shown. Negative value indicates replay strength decreases over time, while positive value means replay increases as a function of sleep time. The statistical inference is done based on a permutation test. The two black dotted lines in each panel indicate the $2.5^{th}$ and $97.5^{th}$ percentile of the permutation samples, respectively. The red solid line indicates the true beta estimate of the effect. Note there is a selection bias in performing statistical inference on forward and backward sequence strength (red rectangle) within this ROI, given the sum of forward and backward sequence is correlated with either forward, or backward sequence alone. There is no selection bias in performing statistics on the difference of sequence effects, or effects relating to time (green rectangle).

In addition to the time varying effect, we can also test the spatial modulation effect, i.e., how replay strength (at the same replay speed) change as a function of its spatial content. For example, is replay stronger for transitions in the start of maze, compared to the end of the track. As an illustrative example, we have used the same ROI defined above, and test the spatial modulation effect on forward replay. Note this test of spatial modulation effect is also unbiased from the overall strength of forward replay, and thereby no selection bias in this ROI, as well.

For visualization purposes, we have first plotted the estimated strength for each pairwise forward sequence (Figure R8a), separately within each scale (from 1 to 4, with increasing spatial scales). The pairwise sequences are ordered from the start of the maze to the end of the maze. Alongside the pair-wise sequence plot, we have plotted the mean replay strength over all possible pairwise transitions (in red), in comparison to the mean of all control transitions (in grey. As expected, they are all around 0). Note that we cannot perform inference on the difference between the red and grey bars here because they have been selected from a biased ROI. It is simply for illustration purposes. We have therefore put them in red squares to match Figure 7f.

To formally test the spatial modulation effect, we can use the exact same approach as outlined above in the **linear contrasts** section. Here, we test a linear increase or decrease across different ***transitions***. We take the linear contrast weight vector, $c$ ([-2,-1,0,1,2] for the largest scale, [-3:3] for the next scale, [-5:5] for the next scale, [-12:12] for the smallest scale) and multiply these by the beta estimates of the transitions:

$$contrast = c^T \beta$$

(17)

If this new measure, $contrast$, is different from zero, then there is a linear increase/decrease from one end of the track to the other. Note that this new contrast is no longer biased by the ROI selection as each transition contributed equally to the ROI selection, but we are now comparing between transitions. Inference on this contrast is therefore valid. We have therefore put them in green boxes to match Figure 7f (Figure 8b, c).

Within the larger two scales, these contrasts are significantly negative (tested against permutations in exactly the same way as the "mean" contrasts). Since we are still in the linear domain, we can now just average these contrasts across the 4 scales and get a single measure for spatial modulation of replay. This average measure is significantly negative (Figure 8b). Hence, on average, forward replay is stronger at the beginning of the track.

We can do the same thing for backward replay. We found a opposite pattern, i.e., strength of backward replay is stronger at the end of the track, and similarly, it is not significant in the smallest scale, and become significant in the largest scale, and also significant on average across all scales (Figure 8c). Again, since we are in the linear domain, we can further contrast these contrasts, asking if this effect is different for forwards and backward replay. We found the difference is indeed significant (Figure 8d). This set of results is consistent with previous rodent literature [60]. Note we would like to stress again, that this analysis is not about a single replay event but is testing for average differences across all replay events.

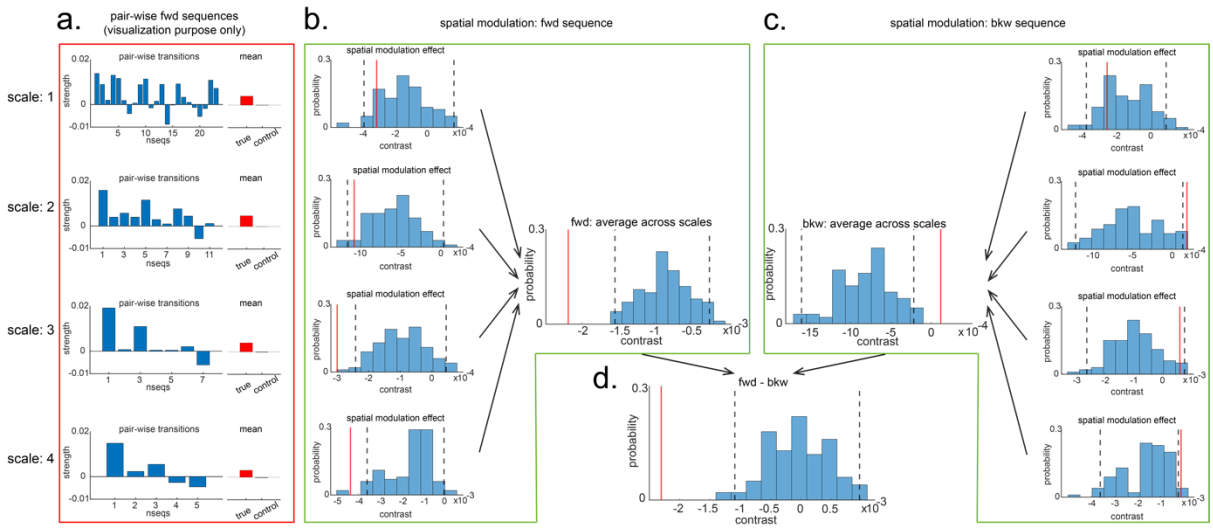



**Figure 8. Pairwise sequence & spatial modulation effect. a.** Within each scale, strengths of each pairwise forward sequences in the ROI (significant replay speeds, cf. Figure R7d, green shading) are ordered from the start of maze to the end of the maze; alongside that, the mean sequence strength across all of these valid pairwise transitions is plotted (red) in comparison to the mean of all control transitions (grey). This is for visualization purpose only and is included in the red rectangle. **b.** The contrast defining a linear change in forward sequenceness across the track (spatial modulation) is shown (red line), both separately for each scale, and average across scales, and compared to permutations. On average, forward replay is stronger at the beginning of the track. **c.** Same as panel b, but this is for the backward sequences. Unlike forward replay, backward replay is stronger at the end of the track. Note, both panel b and c are a about spatial modulation effect, which is orthogonal to overall sequence strength, allowing valid inference. They are therefore included in green boxes. **d.** The difference of this spatial modulation effect between forward and backward sequence is also significant. The black dotted lines indicate the 2.5th and 97.5th percentile of the permutation samples. The red solid line indicates the estimate of the true contrast effect.

Notably, extra care needs to be exercised for second-order questions (compared to first order ones). Problems can emerge due to biases in second order inference, such as in behavioral sampling (e.g., track 1 may be experienced more than track 2 during navigation. This creates a bias when evaluating replay in tack 1 vs. track 2 during rest). Such issues are real but can be finessed by experimental design considerations of a sort commonly applied in the human literature. For example:

(1) Ensure that biases that might occur within subjects will not occur consistently in the same direction *across subjects* (e.g., by randomising stimuli across participants).
(2) Compare across conditions in each subject.
(3) Perform a random effects inference across the population, by comparing against the between-subject variance.

Such approaches are not yet common in rodent electrophysiology and may not be practical in some instances. In such cases, it remains important to be vigilant to guard against these biases with TDLM as with other techniques. If these approaches are feasible, the machinery for computing second-order inferences is straightforward in a linear framework like TDLM.

## GENERALITY OF TDLM

We have now discussed the applicability of TDLM in relation to human MEG, as well as in rodent electrophysiology (with comparisons to standard replay detection methods). A preliminary attempt at detecting replay in human EEG was also shown in the Appendix 4. We believe this establishes TDLM as a domain-general sequence analysis method: TDLM works at the level of decoded state space, rather than the sensor/cell level of the data. It can be applied to a wide range of data types and settings in both humans and rodents, stimulating cross-fertilization across disciplines. It is based on the GLM framework, and this lends it flexibility for regressing out potential confounds while offering an intuitive understanding of the overall approach.

In this section, we discuss the generality of TDLM.

**States:** TDLM assesses the statistical likelihood of certain transitions on a graph. In its original form, TDLM works on discrete states (i.e., nodes in the graph). Continuous spaces can be incorporated by chunking them into discrete spaces. Furthermore, by averaging the same replay speeds measured at multiple scales of discretization (see section “TDLM FOR RODENT REPLAY”), the statistical benefits of an assumption of a Euclidean geometry can be recovered.

**Time length**: The longer the time length, the more accurate the estimates in TDLM. This is because TDLM assesses sequence evidence based on a GLM framework, where time length is the sample size. Higher sample size will lead to more accurate estimates. In the case of rodent analysis, we recommend applying TDLM to aggregated replay events rather than to a single event because this results in 1) more time samples for estimation; 2) more activated states in the analysis time framework. Unlike other techniques which search for a single replay in a single event, this aggregation can be implemented without losing generality, as

TDLM is able to handle multiple sequences in the same data with respect to different directions, contents or speeds. Furthermore, by aggregating linearly across all replay events of the same condition, it provides a natural measure for comparing replay strength, speed and direction across different experimental conditions.

TDLM has already proved important in human experiments where complex state-spaces have been used [22,24,25,35]. We expect this generality will also be important as rodent replay experiments move beyond 1D tracks, for example to foraging in 2D, or in complex mazes.

## DISCUSSION

TDLM is a domain general analysis framework for capturing sequence regularity of neural representations. It is developed on human neuroimaging data, and can be extended to other data sources, including rodent electrophysiology recordings. It offers hope for cross-species investigations on replay (or neural sequences in general), and potentially enable studies of complex tasks in both human and animals, e.g., complex 2D maze in rodents.

TDLM adds a new analysis toolkit to the replay field. It is especially suited for summarising replay strength across many events, for comparing replay strength between conditions, and for analysing replay strength in complex behavioural paradigms. Its linear modelling nature makes it amenable to standard statistical tests, and thereby allows wide use across task, modality, and species. Unlike alternative tools, we have not shown TDLM applied to individual replay events.

The temporal dynamics of neural states have been studied previously with MEG [40,61]. Normally such states are defined by common physiological features (e.g., frequency, functional connectivity) during rest, and termed resting state networks (e.g., default mode network [62]). However, these approaches remain agnostic about the ***content*** of neural activity. The ability to study the temporal dynamics of representational content permits richer investigations into cognitive processes [6], as neural states can be analysed in the context of their roles with respect to a range of cognitive tasks.

Reactivation of neural representations have also been studied previously [63] using approaches similar to the decoding step of TDLM, or multivariate pattern analysis (MVPA) [64]. This has proven fruitful in revealing mnemonic functions [47], understanding sleep[65], and decision-making[66]. However, classification alone does not reveal the rich ***temporal structures*** of reactivation dynamics. We have described the application of TDLM mostly during off-task state in this paper. However, the very same analysis can be applied to on-task data, to test for cued sequential reactivation [22], or sequential decision-making [67]. For example, the ability to detect sequences on-task allows us to tease apart clustered from sequential reactivation, where this may be important for dissociating decision strategies [34] and their individual differences [22,67]. TDLM, therefore, may allow testing of neural predictions from process models such as reinforcement learning during task performance [68], which have proved hard to probe previously in humans [22-25].

In the human neuroimaging domain, we have mainly discussed the application of TDLM with respect to MEG data. In the Appendix 4, we show TDLM also works well with EEG data. This is not surprising given EEG and MEG are effectively measuring the same neural signature, namely local field potential (or associated magnetic field) on the scalp. We do not

have suitable fMRI data to test TDLM. However, related work has suggested it might be possible to measure sequential reactivation using fMRI [69], but particular methodological caveats need to be considered (e.g., a bias from last events due to slow hemodynamic response) [70]. We believe TDLM can deal with this, given it models out non-specific transitions, although further work is needed. In future, we consider 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 computations.

In the rodent electrophysiology domain, we show what TDLM (its multi-scale version) has to offer uniquely compared to existing rodent replay methods. Most importantly, TDLM works on an arbitrary graph and its generality makes replay studies in complex mazes possible. Its linear framework makes the assessment of time varying effect on replay (Figure 7), or other second-order sequence questions straightforward. In future work, a promising direction will be to further separate process noise (e.g., intrinsic variability within sequences) and measurement noise (e.g., noise in MEG recording). This might be achieved by building latent state-space models as have explored recently in rodent community [18,56].

Together, we believe TDLM opens doors for novel investigations of human cognition, including language, sequential planning and inference in non-spatial cognitive tasks [34,35], as well as complicated tasks in rodents, e.g., forging in 2D mazes. TDLM is particularly suited to test specific neural predictions from process models, such as reinforcement learning. We hope TDLM can promote an across species synthesis between experimental and theoretical neuroscience and, in so doing, shed novel light on neural computation.

## METHODS

### Simulating MEG data

We simulate the data so as to be akin to human MEG.

*Task data for obtaining state patterns*

We generate ground truth multivariate patterns (over sensors) of states. We then add random gaussian noise on the ground truth state patterns to form the task data. We train a logistic regression classifier on the task data so as to obtain a decoding model for each of the state patterns. Later we use this decoding 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).

*Rest data for detecting sequences*

First, to imitate temporal autocorrelations and spatial correlations commonly seen in human neuroimaging data, we generate the rest data using an auto-aggressive model with multivariate (over sensors) gaussian noise and add a dependence among sensors. In some simulations, we also add a rhythmic oscillation (e.g., 10Hz).

Second, we inject a sequence of 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 a 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

## Human Replay dataset

### *Task design*

Participants were required to perform a series of tasks with concurrent MEG scanning (see details in Liu, et al. [24]). 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. At 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 a structural code (position and sequence) for the objects.

### *MEG data Acquisition, Pre-processing and Source Reconstruction*

We follow the same procedure that has been reported in Liu, et al. [24]. 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 [71], 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)."

## Rodent Replay dataset

*Data description*

This data is from Ólafsdóttir, et al. [26]. We analyzed one full recording session data (track running for generating rate map, post-running resting for replay detection) from Rat 2192.

*Task description*

In Ólafsdóttir, et al. [26], rats ran multiple laps on a Z maze, and were then placed in a rest enclosure. The two parallel sections of the Z (190 cm each) were connected by a diagonal section (220 cm). Animals were pretrained to run on the track. At the recording session, rats were placed at one end of the Z-track. The ends and corners of the track were baited with sweetened rice to encourage running from one end to the other. In each session rats completed 20 full laps (30–45 min). Following the track session, rats were placed in the rest enclosure for 1.5 hour.

*Preprocessing*

Following Ólafsdóttir, et al. [26], when generating rate maps we excluded data from both the ends and corners because the animals regularly performed non-perambulatory behaviors there. Periods when running speed was less than 3 cm/s were also excluded. Running trajectories were then linearized, dwell time and spikes were binned into 2 cm bins and smoothed with a Gaussian kernel ($\sigma = 5$ bins). We generated rate maps separately for inbound (track1->track2->track3) and outbound (track3->track2->track1) running.

As in Ólafsdóttir, et al. [26], cells recorded in CA1 were classified as place cells if their peak firing field during track running was above 1 Hz and at least 20 cm wide. The candidate replay events were identified based on multi-unit (MU) activity from place cells during rest time. Only periods exceeding the mean rate by 3 stand deviation of MU activity were identified as putative replay events. Events less than 40 ms long or which included activity from less than 15% of the recorded place cell ensemble were rejected.

We analyzed data from one full recording session (track running for generating rate map, post-running resting for replay detection) of Rat 2192 reported in Ólafsdóttir, et al. [26]. Following the procedure described above, we have identified 58 place cells, and 1183 putative replay events. Replay analysis was then performed on the putative replay events, separately for inbound and outbound rate maps.

## Code availability

Source code of TDLM can be found at https://github.com/yunzheliu/TDLM.

**Data availability**

Data are also available at https://github.com/yunzheliu/TDLM.

**Acknowledgement**

We thank Matthew A. Wilson for help with rodent theta sequence analysis. We thank Elliott Wimmer and Toby Wise for helpful discussion and generous sharing of their data. We thank Matt Nour for helpful comments on a previous version of the manuscript. Y.L. is also grateful for the unique opportunity provided by the Brains, Minds & Machines Summer Course. We acknowledge fundings from the Open Research Fund of the State Key Laboratory of Cognitive Neuroscience and Learning to Y.L., Wellcome Trust Investigator Award (098362/Z/12/Z) to R.J.D., Wellcome Trust Senior Research Fellowship (104765/Z/14/Z), and Principal Research Fellowship (219525/Z/19/Z), together with a James S. McDonnell Foundation Award (JSMF220020372), to T.E.J.B; and Wellcome Trust Senior Research Fellowship (212281/Z/18/Z) to C.B. Both Wellcome Centres are supported by core funding from the Wellcome Trust: Wellcome Centre for Integrative Neuroimaging (203139/Z/16/Z), Wellcome Centre for Human Neuroimaging (091593/Z/10/Z). The Max Planck UCL Centre is a joint initiative supported by UCL and the Max Planck Society.

**Appendix 1: Multi-step sequences**

TDLM can be used iteratively. One extension of TDLM of particular interest is: multi-step sequences. It asks about a consistent regularity among multiple states.

So far, we 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, $A \rightarrow B \rightarrow C$.

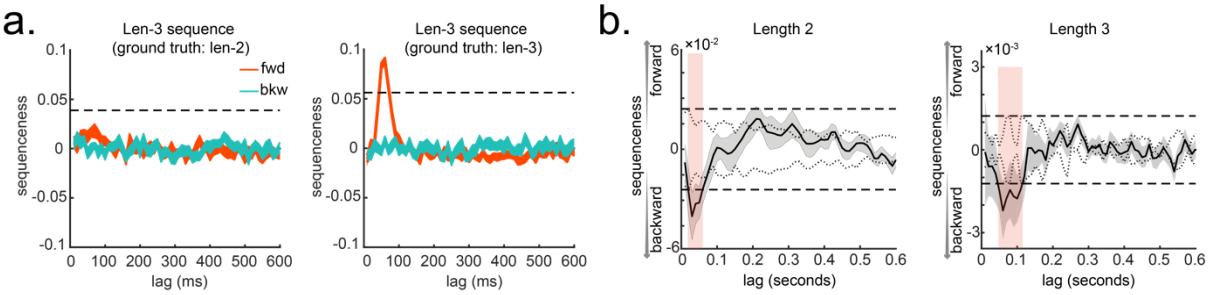
The key ingredient is controlling for shorter sub-sequences (e.g., $A \rightarrow B$ and $B \rightarrow C$), in order to find evidence unique to a multi-step sequence of interest.

Assuming constant state-to-state time lag, $\Delta t$, between A and B, and between B and C. We can create a new state space AB, by shifting B up $\Delta t$, and elementwise multiply it with state A. This new state AB measures the reactivation strength of $A \rightarrow B$, with time lag $\Delta t$. In the same way, we can create a 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 determine the evidence of $A \rightarrow B \rightarrow C$ at time lag $\Delta t$, we can regress AB onto state time course C, at each $\Delta t$ (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 other state but not A. Therefore, similar to Equation 2, we need 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 Appendix 1-figure 1a, see results obtained on human neuroimaging data in Appendix 1-figure 1b). This process can be generalized to any number of steps.

TDLM, in its current form, assumes a constant intra-sequence state-to-state time lag. If there is a variability between state transitions TDLM can still cope, but not very elegantly. Assume 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 then have to test $n$^2 possible time lag combinations. This is a large search space and one that increases exponentially as a function of the length of a sequence.

We note this analysis is different from a typical rodent replay analysis which assesses the overall evidence for a sequence length [16,17]. TDLM asks if there is more evidence for A->B->C, above and beyond evidence for B->C, for example. However, if the main question of interest is "do we have evidence of A->B->C in general", as normally is the case in the rodent replay analysis [16,17], then we should not control for shorter lengths. Instead, we can simply average the evidence together, as implemented in Kurth-Nelson, et al. [35].




**Appendix 1-figure 1. Extension to TDLM: Multi-step 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 positives. **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 reproduced from Figure 3 A, C, Liu et al. 2019, Cell, published under the Creative Commons Attribution 4.0 International Public License (CC BY 4.0).

## Appendix 2: Pseudocode of sensory code and abstract code 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 $a = \frac{1}{2} \sum_{k=1}^{2} a_k$

## Appendix 3: Sequences of sequences

We have detailed the use of either sensory or abstract representations as the states in TDLM. We now take a step further and use sequences themselves as states. Using this kind of hierarchical analysis, we can search for sequences of sequences. This is useful because it can reveal 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 [72] can be cast as sequences of sequences problem: the temporal structure of forward and backward sequence. This can be tested by using TDLM iteratively.

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 $\Delta t$ (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 $\Delta t$). Then, we can elementwise multiply the raw state time course A with the shifted time course B, resulting in a new state AB. Each entry in this new state time course indicates the reactivation strength of sequence AB at a given time.

The general two-level GLMs framework still applies, but now with one important caveat. The new sequence state (e.g., AB) is defined based on the original states (A and B), and where we are now interested in a reactivation regularity, i.e., sequence, between sequences, rather than the original states. We need therefore to 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 including 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 $\Delta t_w$. 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 $\Delta t_w$).

In the first level GLM, TDLM ask for the strength of a 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 $\Delta t$, TDLM estimated a separate linear model:

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

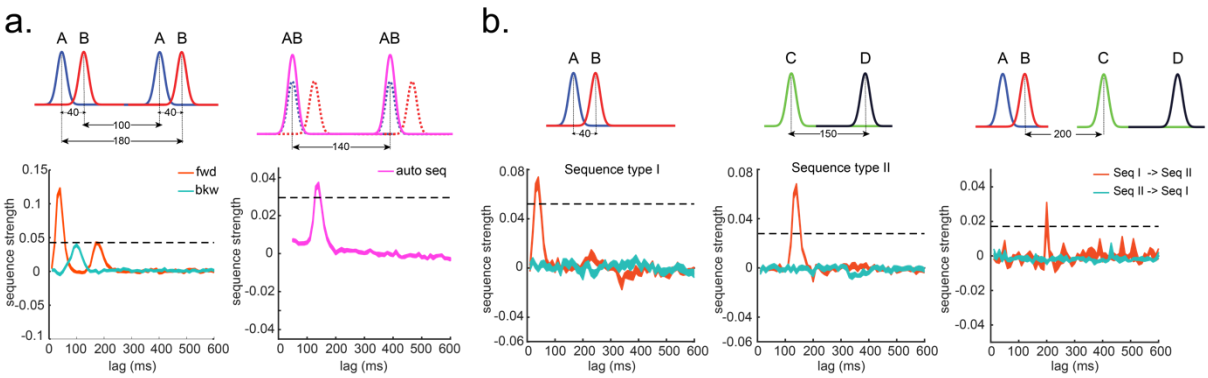
(13)

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

At the 2nd level GLM, TDLM asks how strong the evidence for a 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 (Appendix 3-figure 1a). One interesting application of this is to look for theta sequence [73-75]. One can think of theta sequence, a well-documented phenomenon during rodent spatial navigation, as a neural sequence repeating itself in theta frequency (6 - 12 Hz).

In addition to looking for temporal structure of the same sequence, the method is equally suitable when searching for temporal relationships between different 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 (Appendix 3-figure 1b, left and middle panel); and there is a gap of 200 ms between the two types of sequences (Appendix 3-figure 1b, right panel). These time lags are set arbitrarily for illustration purposes. TDLM can accurately capture the dynamics both within and between the sequences, supporting a potential for uncovering temporal relationships between sequences in general under the same framework.



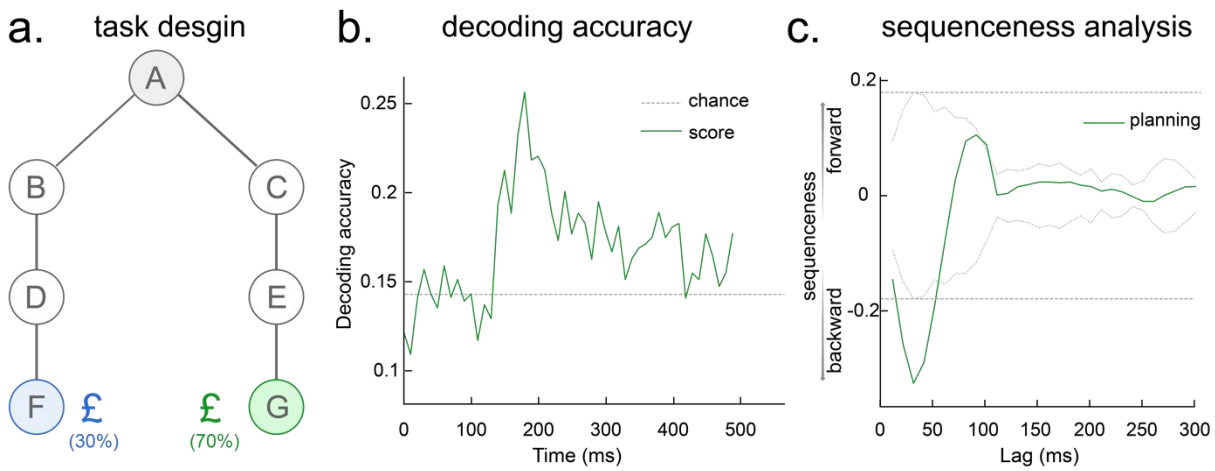

**Appendix 3-figure 1. Sequences of sequences. a,** TDLM can also be used iteratively to capture the repeating pattern of a sequence event itself. Illustration in the top panel describes the ground truth in the simulation. Intra-sequence temporal structure (right) and inter-sequence temporal structure (right) can be extracted simultaneously. **b,** Temporal structure between and within different sequences. Illustration of two sequence types with different state-to-state time lag within sequence, and a systematic gap between the two types of sequences on top. TDLM can capture the temporal structures both within (left and middle panel) and between (right panel) the two sequence types.

## Appendix 4: Apply TDLM to human whole-brain EEG data

An autocorrelation is commonplace in neuroimaging data, including EEG and fMRI. TDLM is designed to specifically take care of this confound and, on this basis, we should be able to work with EEG and fMRI data. We do not have the suitable fMRI data available to test

TDLM but are interested to investigate this in more depth in our future work. We had collected EEG data from one participant to test whether TDLM would *just* work.

The task was designed to examine online sequential replay in online decision-making, by Dr. Toby Wise. This is a 'T-maze' like task, where a participant needs to choose a left or right path based on the value received at the end of the path. We could decode 7 objects well on the whole-brain EEG data using just raw amplitude (same with our MEG-based analysis), and could detect fast backward sequenceness (peaked at 30 ms time lag) during choice/planning time (Appendix 4-figure 1), similar to our previous MEG findings [35]. As this result is from one subject, we are cautious about making an excessive claim, but nevertheless we believe the data show the approach is highly promising for EEG data.

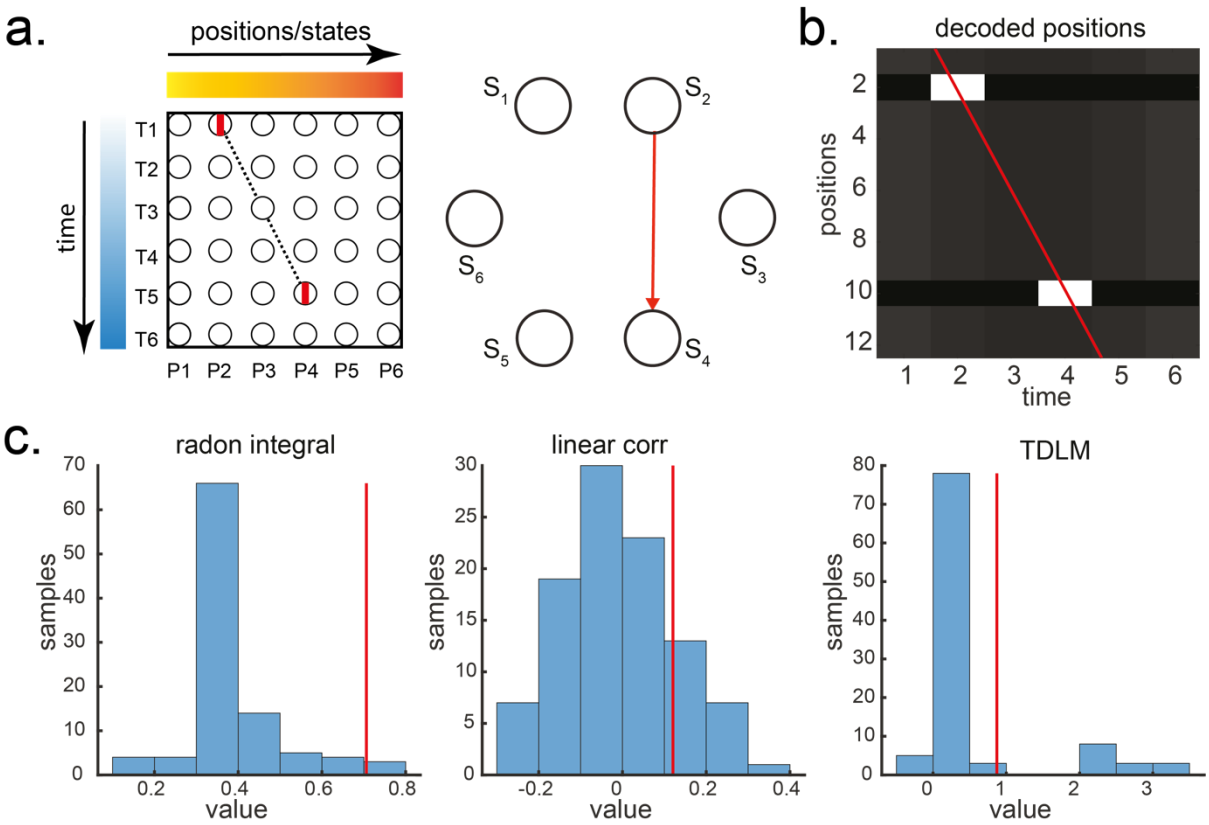



**Appendix 4-figure 1**. **Sequence detection in EEG data (from one participant). a,** Task design. At each trial, the participant starts at state A, and he/she needs to select either "BDF" or "CEG" path, based on the final reward receipt at terminal states F and G. All seven states are indexed by pictures. **b,** The leave-one-out cross-validated decoding accuracy is shown, with a peak at around 200 ms after stimulus onset, similar to our previous MEG findings. **c,** TDLM method is then applied on the decoded state time course where we find a fast backward sequenceness that conforms to task structure. Shown here is a subtraction between forward and backward sequenceness, where a negative sequenceness indicates stronger backward sequence replay. 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, thereby controlling for multiple comparisons. This is the same statistical method used in our previous empirical work, and in the current paper. These EEG sequence results replicate our previous MEG-based findings based on analyses at planning/decision time (see Figure 3 in Kurth-Nelson et al., 2016, and also see Figure 3f in Liu et al., 2019).

## Appendix 5: Less sensitivity of TDLM to skipping sequences

In a linear track where replays only go in a single direction, it is possible that TDLM is less sensitive compared to the linear correlation or the Radon method, given the latter assumes a parametric relationship between space and time. For example, if only the first and last state

are activated, but not the intermediate states, the existing methods **will** report replay, but TDLM **will not**, because in existing methods **space and time are parametric quantities** (Appendix 5-figure 1). In contrast, TDLM only knows about transitions on a graph.




**Appendix 5-figure 1**. **Parametric relationship between space and time vs. graph transitions**. **a,** A scheme for the decoded time by position space is shown (left). Both Radon and linear weighted correlation methods aim to capture a parametric relationship between space and time. TDLM, on the other hand, tries to capture transitions in a graph (shown in right, with the red indicating the transition of interest). **b,** A decoded time by position matrix from simulated spiking data. **c,** Replay analysis using all three methods on this decoded position matrix. TDLM is less sensitive compared to existing "line search" methods, like radon or linear correlation. The red line indicates the true sequence measure from each of these methods. The bar plots are permutation samples by randomly shuffling the rate maps.

## REFERENCES


1 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).

2 Kriegeskorte, N., Mur, M. & Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience* **2**, 4 (2008).
3 Barron, H. C., Garvert, M. M. & Behrens, T. E. Repetition suppression: a means to index neural representations using BOLD? *Philosophical Transactions of the Royal Society B: Biological Sciences* **371**, 20150355 (2016).
4 Smith, S. M. *et al.* Correspondence of the brain's functional architecture during activation and rest. *Proceedings of the National Academy of Sciences* **106**, 13040-13045 (2009).
5 Tavor, I. *et al.* Task-free MRI predicts individual differences in brain activity during task performance. *Science* **352**, 216-220 (2016).
6 Higgins, C. *et al.* Replay bursts in humans coincide with activation of the default mode and parietal alpha networks. *Neuron* **109**, 882-893. e887 (2021).
7 Wilson, M. A. & McNaughton, B. L. Reactivation of hippocampal ensemble memories during sleep. *Science* **265**, 676-679 (1994).
8 Skaggs, W. E. & McNaughton, B. L. Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science* **271**, 1870-1873 (1996).
9 Louie, K. & Wilson, M. A. Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron* **29**, 145-156 (2001).
10 Lee, A. K. & Wilson, M. A. Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron* **36**, 1183-1194 (2002).
11 Foster, D. J. Replay comes of age. *Annual Review of Neuroscience* **40**, 581-602 (2017).
12 Ólafsdóttir, H. F., Bush, D. & Barry, C. The Role of Hippocampal Replay in Memory and Planning. *Current Biology* **28**, R37-R50 (2018).
13 Pfeiffer, B. E. The content of hippocampal “replay”. *Hippocampus* **30**, 6-18 (2020).
14 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).
15 Lisman, J. *et al.* Viewpoints: how the hippocampus contributes to memory, navigation and cognition. *Nature Neuroscience* **20**, 1434-1447 (2017).
16 Davidson, T. J., Kloosterman, F. & Wilson, M. A. Hippocampal replay of extended experience. *Neuron* **63**, 497-507 (2009).
17 Grosmark, A. D. & Buzsáki, G. Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences. *Science* **351**, 1440-1443 (2016).
18 Maboudi, K. *et al.* Uncovering temporal structure in hippocampal output patterns. *eLife* **7**, e34467 (2018).
19 van der Meer, M. A., Kemere, C. & Diba, K. Progress and issues in second-order analysis of hippocampal replay. *Philosophical Transactions of the Royal Society B: Biological Sciences* **375**, 20190238 (2020).
20 Tingley, D. & Peyrache, A. On the methods for reactivation and replay analysis. *Philosophical Transactions of the Royal Society B: Biological Sciences* **375**, 20190231 (2020).
21 Rosenberg, M., Zhang, T., Perona, P. & Meister, M. Mice in a labyrinth: Rapid learning, sudden insight, and efficient exploration. *bioRxiv* (2021).
22 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* **23**, 1025–1033 (2020).

23 Nour, M. M., Liu, Y., Arumuham, A., Kurth-Nelson, Z. & Dolan, R. Impaired neural replay of inferred relationships in schizophrenia. *Cell* **in press** (2021).
24 Liu, Y., Dolan, R. J., Kurth-Nelson, Z. & Behrens, T. E. J. Human replay spontaneously reorganizes experience. *Cell* **178**, 640-652 (2019).
25 Liu, Y., Mattar, M. G., Behrens, T. E., Daw, N. D. & Dolan, R. J. Experience replay is associated with efficient nonlocal learning. *Science* **372** (2021).
26 Ólafsdóttir, H. F., Carpenter, F. & Barry, C. Coordinated grid and place cell replay during rest. *Nature Neuroscience* **19**, 792 (2016).
27 Barron, H. C., Mars, R. B., Dupret, D., Lerch, J. P. & Sampaio-Baptista, C. Cross-species neuroscience: closing the explanatory gap. *Philosophical Transactions of the Royal Society B: Biological Sciences* **376**, 20190633 (2021).
28 Kurth-Nelson, Z., Barnes, G., Sejdinovic, D., Dolan, R. & Dayan, P. Temporal structure in associative retrieval. *eLife* **4**, e04919 (2015).
29 Fyhn, M., Hafting, T., Treves, A., Moser, M.-B. & Moser, E. I. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature* **446**, 190 (2007).
30 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).
31 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).
32 Deodatis, G. & Shinozuka, M. Auto-regressive model for nonstationary stochastic processes. *Journal of engineering mechanics* **114**, 1995-2012 (1988).
33 Eichler, M. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics* **137**, 334-353 (2007).
34 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).
35 Kurth-Nelson, Z., Economides, M., Dolan, Raymond J. & Dayan, P. Fast Sequences of Non-spatial State Representations in Humans. *Neuron* **91**, 194-204 (2016).
36 Lubenov, E. V. & Siapas, A. G. Hippocampal theta oscillations are travelling waves. *Nature* **459**, 534-539 (2009).
37 Wilson, H. R., Blake, R. & Lee, S.-H. Dynamics of travelling waves in visual perception. *Nature* **412**, 907-910 (2001).
38 Weinberger, K. Q., Blitzer, J. & Saul, L. K. in *Advances in neural information processing systems.* 1473-1480.
39 Higgins, C. *Uncovering temporal structure in neural data with statistical machine learning models*, University of Oxford, (2019).
40 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).
41 Harris, K. D. Nonsense correlations in neuroscience. *bioRxiv* (2020).
42 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).
43 Nichols, T. E. Multiple testing corrections, nonparametric methods, and random field theory. *Neuroimage* **62**, 811-815 (2012).
44 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).

45 Sakai, K. & Miyashita, Y. Neural organization for the long-term memory of paired associates. *Nature* **354**, 152-155 (1991).
46 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).
47 Wimmer, G. E. & Shohamy, D. Preference by association: how memory mechanisms in the hippocampus bias decisions. *Science* **338**, 270-273 (2012).
48 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).
49 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).
50 Sirota, A. *et al.* Entrainment of neocortical neurons and gamma oscillations by the hippocampal theta rhythm. *Neuron* **60**, 683-697 (2008).
51 Buzsáki, G. & Vanderwolf, C. H. Cellular bases of hippocampal EEG in the behaving rat. *Brain Research Reviews* **6**, 139-171 (1983).
52 Hincapié, A.-S. *et al.* The impact of MEG source reconstruction method on source-space connectivity estimation: a comparison between minimum-norm solution and beamforming. *Neuroimage* **156**, 29-42 (2017).
53 O'Neill, G. C. *et al.* Testing covariance models for MEG source reconstruction of hippocampal activity. *bioRxiv* (2021).
54 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).
55 Toft, P. A. The Radon transform-theory and implementation. (1996).
56 Denovellis, E. L. *et al.* Hippocampal replay of experience at real-world speeds. *bioRxiv* (2020).
57 Karlsson, M. P. & Frank, L. M. Awake replay of remote experiences in the hippocampus. *Nature Neuroscience* **12**, 913 (2009).
58 Nádasdy, Z., Hirase, H., Czurkó, A., Csicsvari, J. & Buzsáki, G. Replay and time compression of recurring spike sequences in the hippocampus. *Journal of Neuroscience* **19**, 9497-9507 (1999).
59 Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S. & Baker, C. I. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience* **12**, 535 (2009).
60 Diba, K. & Buzsáki, G. Forward and reverse hippocampal place-cell sequences during ripples. *Nature Neuroscience* **10**, 1241 (2007).
61 Baker, A. P. *et al.* Fast transient networks in spontaneous human brain activity. *eLife* **3**, e01867 (2014).
62 Raichle, M. E. *et al.* A default mode of brain function. *Proceedings of the National Academy of Sciences* **98**, 676-682 (2001).
63 Tambini, A. & Davachi, L. Awake Reactivation of Prior Experiences Consolidates Memories and Biases Cognition. *Trends in Cognitive Sciences* (2019).
64 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).
65 Lewis, P. A. & Durrant, S. J. Overlapping memory replay during sleep builds cognitive schemata. *Trends in Cognitive Sciences* **15**, 343-351 (2011).
66 Schuck, Nicolas W., Cai, Ming B., Wilson, Robert C. & Niv, Y. Human Orbitofrontal Cortex Represents a Cognitive Map of State Space. *Neuron* **91**, 1402-1412 (2016).

67 Eldar, E., Lièvre, G., Dayan, P. & Dolan, R. J. The roles of online and offline replay in planning. *eLife* **9**, e56911 (2020).
68 Dayan, P. & Daw, N. D. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience* **8**, 429-453 (2008).
69 Schuck, N. W. & Niv, Y. Sequential replay of nonspatial task states in the human hippocampus. *Science* **364**, eaaw5181 (2019).
70 Wittkuhn, L. & Schuck, N. W. Dynamics of fMRI patterns reflect sub-second activation sequences and reveal replay in human visual cortex. *Nature Communications* **12**, 1-22 (2021).
71 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).
72 Penny, W. D., Zeidman, P. & Burgess, N. Forward and backward inference in spatial cognition. *PLoS computational biology* **9** (2013).
73 Mehta, M., Lee, A. & Wilson, M. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature* **417**, 741 (2002).
74 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).
75 Buzsáki, G. & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature Neuroscience* **16**, 130 (2013).