Marked point process filter for clusterless and adaptive encoding-decoding of multiunit activity

Kensuke Arai[1]

[1]Boston University Department of Mathematics and Statistics, Boston University, Boston MA 02215, U.S.A.

Daniel F. Liu[1]

[1]University of California BerkeleyUniversity of California San Francisco Graduate Program in Bioengineering, UCSF, San Francisco, CA 94158, U.S.A.

Loren M. Frank[1]

[1]Department of Physiology, University of California San Francisco, San Francisco, CA 94158, U.S.A.

Uri T. Eden[1]

[1]Department of Mathematics and Statistics, Boston University, Boston, MA 02215, U.S.A.

Keywords: Marked point process, hippocampus, real-time algorithm, adaptive

Abstract

Real-time, closed-loop experiments can uncover causal relationships between specific neural activity and behavior. An important advance in realizing this is the marked point process filtering framework which utilizes the ”mark” or the waveform features of unsorted spikes, to construct a relationship between these features and behavior, which we call the encoding model. This relationship is not fixed, because learning changes coding properties of individual neurons, and electrodes can physically move during the experiment, changing waveform characteristics. We introduce a sequential, Bayesian encoding model which allows incorporation of new information on the fly to adapt the model in real time. A possible application of this framework is to the decoding of the contents of hippocampal ripples in rats exploring a maze. During physical exploration, we observe the marks and positions at which they occur, to update the encoding model, which is employed to decode contents of ripples when rats stop moving, and switch back to updating the model once the rat starts moving again.

Introduction

Neuroscience experiments are increasingly focused on the coordinated activity of neural populations and how it represents information about biological and behavioral signals. Statistical models of the population neural code allow information about these biological or behavioral signals to be inferred from the simultaneously observed spiking activity of the population. Neural population activity is typically observed extracellularly, with electrodes picking up multi-unit activity whose neural identities are unknown *a priori*. Knowing or estimating the neural identity of spikes is generally important for understanding the coding properties of individual neurons, but there has been increasing interest in statistical models of the coding properties of populations that do not require a spike estimation, or sorting, step.

If the nearby neurons code for similar features [1], a simple statisical model may be created directly from unsorted multi-unit activity that can decode nearly as well as models using sorted single units [2, 1, 3, 4, 5, 6], making real-time decoding with spikes feasible. However, in cases where nearby neurons do not encode similar features, simple models based on multi-unit activity perform poorly compared to separate models for each neuron, requiring a computationally expensive spike sorting step [7], before fitting a model that collectively describe the population activity. For example, hippocampal activity recorded on a single tetrode will often yield multiple place cells whose place fields do not overlap at all, requiring spike sorting to extract spatial information from population data. New experimental designs that require closed-loop estimation and stimulation necessitates a new approach toward modeling population activity in a real-time setting.

The point process framework has been used extensively to model the activity of individual neurons [8, 9, 10, 11] once spikes have been sorted, requiring two independent steps. For population activity, modeling each spike train as separate point processes and including spiking history of other neurons in the population as covariates allows the modeling of the dependencies in the population [11, 12, 13], though cannot account for biologically relevant instantaneous joint spiking events [14, 15, 16]. An extension of the point process framework being increasingly applied to neural data analysis, the marked point process model [17], which can account for instantaneous joint spiking events [18], and has also allowed characterization of spikes according to waveform and the modeling of the populations to be done in single, integrated step. This approach, sometimes known as clusterless encoding and decoding [19, 20, 21, 22], where a joint model for the firing intensity dependent on waveform and stimulus features, behavioral variables or in hippocampal recordings, spatial position, the joint mark intensity function (JMIF) is estimated simultaneously in one step, allowing a streamlined real-time analysis pipeline.

While spike sorting methods have also improved in speed and accuracy [23], there are other advantages the clusterless method might have for modeling and decoding applications. Sorting introduces decoding errors when neural identities are determined by hard decision boundaries, and spikes with ambiguous waveforms may be classified incorrectly, in contrast to probabilistic assignment of spikes [24, 25, 19, 20]. Ambiguous waveforms in the clusterless method are automatically less informative to the decoder [20], making them less likely to cause large errors, and more suited to real-time decoding applications.

However, there are still practical issues which must be addressed with the clusterless method. First, the class of models typically used for the JMIF have been non-parameteric, resulting in an ever-increasing model size as new data becomes available. When tens to hundreds of thousands of spikes from a population are available, the computational cost of computing the model estimates may preclude real-time and closed-loop analyses. Second, neural firing properties are generally non-stationary; this may be due to changes in the spike waveforms from individual neural due to mechanical instability of electrodes or changes in background noise statistics, or due to biologically relevant changes in tuning properties of individual neurons or the population as with learning and plasticity [26].

For models of sorted spike train, a number of approaches for these issues have been addressed in part by spike sorting and clusterless algorithms. In online, real-time settings, spiking and behavioral data arrive as they become available, and non-parametric estimation methods like the kernel density estimate, linearly increase in size with additional data.

Sodkomkham et al [21] have added heuristic rules for merging new spikes to existing clusters in the position-mark space based on the Mahalanobis distance between them to keep model size from growing as more spikes arrive, making decoding possible in real time even when training data is large. However, the fitting procedure treats space and waveform features equally without accounting for nonuniform sampling of space, making accommodation for nonstationarity of the JMIF more challenging. Nonstationarity of spike waveforms due to the mean waveform changing over longer timescales, possibly due to sometimes abrupt electrode drift and/or biological or background noise changes [26], can cause sometimes drastic changes in the statistics of the signal. Spike sorting algorithms account for nonstationarity of waveform features and can accommodate appearance and disappearance of neurons [27, 28, 29], but do not support online update of the model. Nonstationarity manifests also in non-spike sorted brain computer interface applications, and investigators have automated the recalibration by taking advantage of the relatively invariant low-dimensional manifold for which the neural spiking noisely codes for [5] or by taking advantage of very long histories of neural-to-kinematic mappings by using nonlinear multiplicative recursive neural networks [6]. Our contribution, the joint mark intensity function is sequentially updated mixture of Gaussians (JOYFULMoGs), builds on the marked point process framework by addressing these disparate issues using an adaptive mixture of Gaussians structure for the JMIF model, with time-varying parameters estimated under a comprehensive Bayesian framework. The mixture of Gaussians is a parsimonious choice for the model class, which requires much fewer parameters than an alternative model class such as multidimensional splines, particularly with each additional mark and spatial dimension. One application area for JOYFULMoGs is for real-time estimation of hippocampal replay events during navigation tasks. Hippocampal replay involves the reactivation of patterns of spiking activity during rest periods that occurred previously during active exploration, and the informational content of this activity likely changes throughout the course of an experiment, necessitating the need for models that can be estimated, updated and adapted in a computationally efficient manner in real time. Under JOYFULMoGs, the encoding model would be updated through each period of active exploration, and the resulting model would be used to estimate replay content during the rest periods. Note that these rest periods could be periods of sleep between experimental epochs or they could be brief pauses between exploration within an experimental epoch.

While building the full real-time decoder of hippocampal replay and deploying it in experiment is beyond the scope of this paper, we illustrate the application of JOYFULMoGs to real offline data from rat hippocampus by addressing a simpler problem to illustrate the framework. We will divide an extended period of active exploration into a sequence of shorter epochs, and alternate between adapting the encoding model in the first half of the epoch, and using that model to decode the actual movement of the rat in the second half of the epoch. The halves need not be of equal temporal duration. We call the nth encoding/decoding epoch I_n, which contains an encoding $\mathrm{I}^{\mathrm{e}}_n$, followed by a decoding $\mathrm{I}^{\mathrm{d}}_n$, period. Both the spike waveforms (the marks) and behavior observed during $\mathrm{I}^{\mathrm{e}}_n$ are used to update the model at the end of $\mathrm{I}^{\mathrm{e}}_n$, and the marks observed from during $\mathrm{I}^{\mathrm{d}}_n$ will be decoded at the end of $\mathrm{I}^{\mathrm{d}}_n$ and compared with actual movements the animal made. Our model is able to handle ”appearance” and ”disappearance” of neurons and receptive fields, as well as drifts in receptive field location, firing rate and tuning widths and changes to the spike waveforms. The estimated posterior from $\mathrm{I}^{\mathrm{e}}_n$ is incorporated as a prior for the same parameters in $\mathrm{I}^{\mathrm{e}}_{n+1}$. Decoding with this model is computationally inexpensive as there are only a small number of parameters, and model size is kept constant even as new data arrives. Uncertainty can be added to the posterior distribution when it is carried over as a prior in to the next $\mathrm{I}^{\mathrm{e}}_{n+1}$, to accommodate statistical changes in receptive fields and recording quality as the experiment progresses [30, 31].

The paper is organized as follows. In the Methods section, we motivate sequential updating of the encoding model, and introduce notation used throughout the paper. We then describe the 2-step Gibbs sampling procedure, and derive the conditional posterior distributions for each of the model parameters. We also describe how uncertainty is added to the estimated posteriors when they are used as priors for the next epoch, and also how the number of clusters is updated. After describing the model, we describe a simulation of a rat running though a circular maze and simulated place field activity, before demonstrating JOYFULMoGs on dynamic simulated place fields and on actual experimental data, comparing how our method works in data settings where the data is likely to be very dynamic with changing neural representation during the experiment, and a setting where the representation is not expected to change as much.

Methods

Real-time decoders of neural activity must incorporate continually arriving new data into the encoding model, as refitting the model from scratch every time new data arrives would be computationally expensive. Further, during the experiment, changes to the neural activity might cause some features of the encoding properties to disappear, while new features may appear, and signal and noise statistics of the spike waveforms on the electrodes may change as electrodes physically move. Fitting without regards to time would overestimate variability in the signal due to these sorts of longer-timescale trends. We propose to address these two concerns using sequential Bayesian updates to the JMIF. Since the experimental design involves interleaved periods of encoding and decoding, instead of updating with each new spike, we accumulate spikes over a period of continuous physical movement. All of the data that have arrived by the nth epoch, $Y^n = \{Y_1, Y_2, ...Y_n\}$, come in packets, and at the end of each $\mathrm{I}^{\mathrm{e}}_n$, we combine the likelihood from the spikes and trajectory, Y_n, observed during $\mathrm{I}^{\mathrm{e}}_n$, with the summary of the current state of the encoding model, the posterior from $\mathrm{I}^{\mathrm{e}}_{n-1}$, $p(\Theta|Y^{n-1})$, interpreted as a prior, to sequentially update the encoding model, Fig. 1. The posterior distribution at the end of $\mathrm{I}^{\mathrm{e}}_n$ is then, by Bayes Rule,

$$p(\Theta|Y) \propto p(Y_n|\Theta, Y^{n-1})p(Y^{n-1}|\Theta)p(\Theta) \propto p(Y_n|\Theta)p(\Theta|Y^{n-1}). \tag{1}$$

New data is combined with the prior knowledge $p(\Theta)$ for the first epoch or $p(Y^n|\Theta)p(\Theta)$ for the nth epoch, allowing a continual update of our estimate of Θ by incorporating new data Y as it becomes available in real time. The posterior from $\mathrm{I}^{\mathrm{e}}_{n-1}$ may be used as a prior for $\mathrm{I}^{\mathrm{e}}_n$ without change, or we may systematically add variance to this prior to keep the prior variance from becoming so small that new evidence provided by the most recent spikes, fails to allow the model to follow longer time-scale non-stationarities in the data.

We seek to characterize the place-specific firing of electrophysiologically recorded cells on K channels from rat hippocampus while it explores a maze with a trajectory x_t. Because individual neurons with 1 or more distinct place fields presumably generated the neural activity, position and K-dimensional mark of spikes often appear clustered when plotted simultaneously, Fig. 2. Previous work used a non-parametric kernel density estimation to estimate this $\lambda(t, \mathbf{m})$ [19, 20]. Here, we approximate it as a mixture of C Gaussians $\lambda(t, \mathbf{m}) = \sum_c^C \lambda_c(t, \mathbf{m})$, with Gaussian components

$$\lambda_c(t, \mathbf{m}) = \frac{\lambda_{0,c}}{\sqrt{(2\pi)^{K+1}\sigma_c^2|\mathbf{\Sigma}_c|}} \exp\left[-\frac{(x_t - f_c)^2}{2\sigma_c^2} - \frac{1}{2}(\mathbf{m} - \boldsymbol{\mu}_c)^T \mathbf{\Sigma}_c^{-1}(\mathbf{m} - \boldsymbol{\mu}_c)\right]. \quad (2)$$

The hyperparameters for the centers are μ_{f_c}, $\sigma_{f_c}^2$, $\boldsymbol{\mu}_{\boldsymbol{\mu}_c}$, and $\Sigma_{\boldsymbol{\mu}_c}$, for the widths are $\alpha_{\sigma_c^2}$, $\beta_{\sigma_c^2}$, $\nu_{\mathbf{\Sigma}_c}$ and $\mathbf{\Psi}_{\mathbf{\Sigma}_c}$, and those for the intensities are $\alpha_{\lambda_{0,c}}$ and $\beta_{\lambda_{0,c}}$

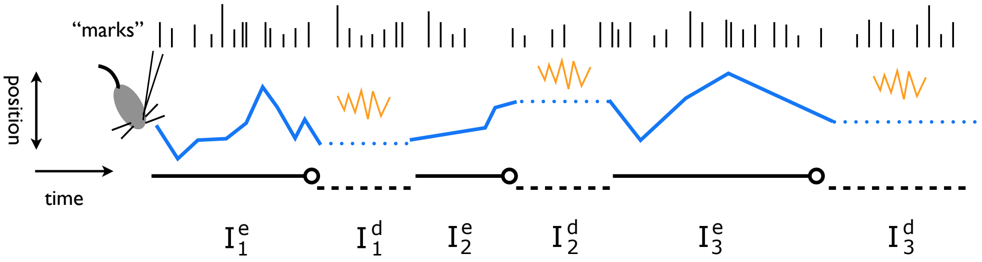

Figure 1: Schematic of experimental design with alternating episodes of encoding and decoding. Ticks at the top represent unsorted spikes or ”marks”, whose varying lengths represent a waveform feature, e.g. spike amplitude. Below the marks, trajectory in blue is plotted, with solid lines representing physical movement, and dotted lines representing physical or replayed movement. Straight black lines beneath the trajectory represent time periods. Initial encoding model is created at the time point represented by the open circle, with prior information provided by the experimenter, using accumulated spikes and physical trajectory observed in $\mathrm{I}_1^{\mathrm{e}}$, Y_1. Movements or contents of hippocampal replays (sharp wave ripple as orange squiggle) occurring during $\mathrm{I}_1^{\mathrm{d}}$ can be decoded by plugging the spikes occurring during replay into the initial encoding model. When the animal starts moving again during $\mathrm{I}_2^{\mathrm{e}}$, once again, spikes and physical trajectory observed during $\mathrm{I}_2^{\mathrm{e}}$, Y_2, are used to the update the initial encoding model at the time point represented by the open circle, by using the initial model, $p(\Theta|Y^1)$ as a prior to be combined with the likelihood of the data observed during $\mathrm{I}_2^{\mathrm{e}}$. The encoding model is used again to decode movements or hippocampal replay contents occuring during $\mathrm{I}_2^{\mathrm{d}}$.

Fitting procedure: Gibbs sampling

In $\mathrm{I}_n^{\mathrm{e}}$, we observe $Y_n = \{\Delta\mathrm{N}, \mathbf{M}, \mathbf{x}\}$ in T discrete time bins. $\Delta\mathrm{N}$ is a vector of size T indicating how many spikes ≥ 0, are observed in each of the T time bins. $\mathbf{M}$ is a vector of size T, whose tth entry is a matrix of size $\Delta N_t \times K$ containing the $\Delta N_t \geq 0$ marks of dimension K of the observed spikes. $\mathbf{x}$ is a vector of size T, and is the trajectory of the animal during this epoch. Initially when $n = 0$, a guess is made as to the number M_G and locations of (spatial-mark) clusters that are present in the data, to initialize the Gibbs sampler. We initialize with M components slightly larger than M_G, and for the clusters with indices $> M_G$, we randomly set their centers and widths to be random values. For the other components c, we use sample statistics for Θ_c as starting parameter values. However, we leave the priors for all the Θ broad to express our uncertainty in their values.

The Gibbs sampling involves repeating two steps: the stochastic allocation of all spikes into clusters given current parameter values, followed by the conditionally independent sampling of the parameters of each cluster given the assignments in the previous step [32]. Typically in Gibbs sampling in a model with parameters Θ and data Y, the $\theta \in \Theta$ are sampled sequentially in each Gibbs iteration, from the conditional posterior $p(\theta|Y, \Theta_{\backslash\theta})$. The conditional posterior is obtained by combining the likelihood $p(Y|\theta, \Theta_{\backslash\theta})$ with all $\Theta_{\backslash\theta}$ regarded as constant, and a prior conjugate to this likelihood. After a large number of iterations (we choose 15,000) to sample an empirical posterior distribution, model parameter values are commonly read off using the empirical mean, mode or median. Here we choose the empirical median. JOYFULMoGs then uses the empirical posterior distribution as a prior for Θ when updating the model with new data in the next epoch. In order to do this, we fit the marginal empirical distributions of each parameter θ with the distribution conjugate to the likelihood with all $\Theta_{\backslash\theta}$ constant, using maximum likelihood to find its hyperparameters. This approximation of the posterior distribution by marginalization is reasonable, because when samples of a parameter pair (x, y) are examined from the marginalized empirical distribution

$p(x, y) = \int p(\Theta) d\Theta_{\setminus(x,y)}$), the samples often appear to have little correlation, suggesting that the joint distribution of the parameters Θ are nearly independent and the product of the marginal posteriors should be a good approximation to the joint posterior distribution[1]. We also note that not all of the parameters used in mixture of Gaussians JMIF have a likelihood with such a conjugate prior (notably f_c and $\sqrt{\sigma_c^2}$). In those cases, we explain our choice of prior in the following section.

Stochastic allocation into clusters

Conditioned on fixed parameter values Θ, a stochastic assignment of $S = \sum_{t=0}^{T} \Delta N_t$ observed marks into C clusters is performed. Because more than 1 spike may be observed in a time bin, we introduce notation $\#(t, w)$, which maps the time bin number and the spike index within that bin to a single integer index $\in [0, S]$. A single row of the indicator matrix $\mathbf{Z}$ has 0s in $C - 1$ columns, and a 1 in the column corresponding to the cluster to which this mark belongs, so $S_c = \sum_t \sum_w^{\Delta N_t} z_{\#(t,w),j=c}$ is the number of marks assigned to cluster c. The probability of assignment of the wth mark in time bin t to the 1st cluster is

$$p(\mathbf{z}_{\#(t,w)} = [1, 0, 0...]|\Theta, x_t, \mathbf{m}_{t,w}) = \frac{\lambda_{0,1} N(x_t, \mathbf{m}_{t,w}|f_1, \sigma_1^2, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\sum \lambda_{0,j} N(x_t, \mathbf{m}_{t,w}|f_j, \sigma_j^2, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{3}$$

[1] A notable exception occurs when a cluster's $\sqrt{\sigma_c^2}$ is wider than the width of the sampled trajectory, W. σ_c^2 strongly constrains the conditional posterior for λ_c, yet for widths much larger than W, the shape of the cluster in that observable space is barely affected by any particular choice of value of $\sqrt{\sigma_c^2} \gg W$, inducing a strong correlation between the samples of $\sqrt{\sigma_c^2}$ and $lambda_{0,c}$. However, most informative clusters have $\sqrt{\sigma_c^2} \ll W$ than the whole of sampled space, and this correlation with the intensity is not observed.

Once **z** is assigned, fitting the parameters of each individual cluster can proceed independently of all other clusters.

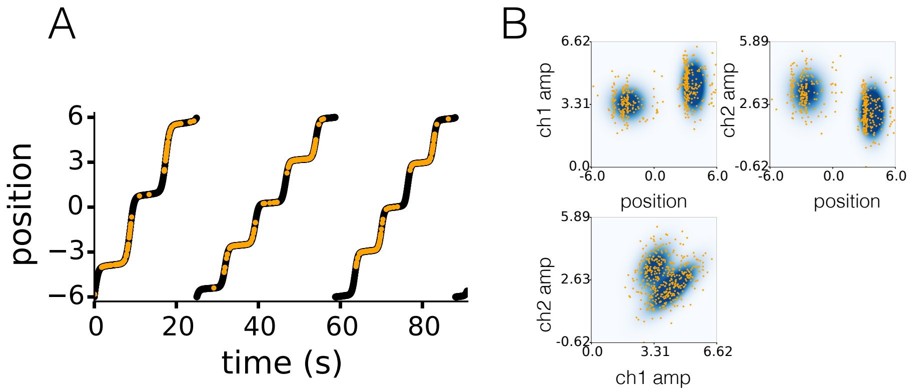

Figure 2: A) The trajectory of a simulated rat over a circular track, x parameterized from $[-6 : 6]$. Orange dots overlayed are the time and location where an unsorted spike was observed. Rat traverses track in stops and starts. B) Top 2 rows, position vs. channel 1 or 2 and bottom row, channel 1 vs. channel 2 plot of observed marks plotted on top of marginalized JMIF. Note that the center of the scattered marks does not coincide with the fitted clusters due to the non-uniformity of the trajectory.

Sampling the parameters Θ

Conditioned on the assignments $\mathbf{Z}$ made in the previous step, parameters Θ are sampled. The likelihood of observing ΔN_t marks at time t is

$$P(\Delta N_t, \mathbf{m}_t | x_t, \Theta, \mathbf{Z}) \propto \begin{cases} \exp[-\Delta\Lambda(x_t)], & \Delta N_t = 0 \\ \prod_i^{\Delta N_t} \lambda_c(x_t, \mathbf{m}_{t,i}) \Delta \exp[-\Delta\Lambda(x_t)] & \Delta N_t \geq 1 \end{cases} \tag{4}$$

where

$$\Lambda(x_t) = \int \lambda(x_t, \mathbf{m}) d\mathbf{m} = \sum_c \Lambda_c(x_t) = \sum_c \int \lambda_c(x_t, \mathbf{m}) d\mathbf{m} \tag{5}$$

is the marginalized JMIF. We wish to estimate the parameters $\Theta = \{\lambda_{0,c}, f_c, \sigma_c^2, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$. The likelihood of observing the marks $\{\Delta\mathbf{N}, \mathbf{M}\}$ given the trajectory x_t and cluster assignments $\mathbf{Z}$ is

$$\begin{aligned} P(\Delta\mathbf{N}, \mathbf{M}|\Theta, \mathbf{x}, \mathbf{Z}) &\propto \left[\prod_{b=1}^{B_S} \prod_{w=1}^{\Delta N_{s_b}} \Delta\lambda(x_{s_b}, \mathbf{m}_{b_w})\right] \prod_{j=1}^{T} \exp[-\Delta\Lambda(x_j)] \\ &\propto \left[\prod_{b=1}^{B_S} \prod_{w=1}^{\Delta N_{s_b}} \sum_{z_{\#(b,w)}} \prod_{c} [\lambda_c(x_{s_b}, \mathbf{m})]^{z_{\#(b,w),c}}\right] \prod_{c} \exp\left[-\Delta \sum_{j}^{T} \Lambda_c(x_j)\right]. \end{aligned} \tag{6}$$

For the last exponential term, we exchange order of the product of terms evaluated at multiple timepoints (and positions) with the sum over all the Gaussian components, to obtain the last line. Rewritten, we see each component in the product can be approximated as $\exp\left[-\Delta \sum_j^T \Lambda_c(x_j)\right] \approx \exp\left[-\Delta \sum_i N(x_i|f_c, \sigma_c^2)h_i\right]^{\lambda_{0,c}}$, whose argument is a roughly a convolution of a Gaussian with mean f_c and variance σ_c^2 with the spatial occupation density. Convolutions are expensive to calculate at every Gibbs iteration, but this simple rearrangement allows us to approximate this term by calculating the convolution over a 2-dimensional grid of means and variances once before the start of Gibbs sampling, and doing lookups for specific values of f and σ^2. To emphasize our computational approach, we write this term as $\exp\left[-V_e(f_c, \sigma_c^2)\right]^{\lambda_{0,c}}$, with the e emphasizing that, while this function changes each epoch, its evaluation is generic across all C clusters once the path traversed in the epoch is given. This term takes into account the nonuniform animal trajectory, Fig. 2. Absent this term, if the animal moved very slowly through a region on the outskirts of a place field of a place cell, but quickly moved through the center of a place field of the same cell, the estimate of the place field center would be biased to the outskirts. The $\mathbf{z}_{\#(b,w)}$ in the exponent in Eq. 6 groups terms by cluster assignment of the spikes, so that the expressions for the C conditional likelihoods for each cluster only involve spikes assigned to those clusters. Thus once an assignment is made, the sampling for each of the C cluster parameters can proceed independently.

Samples for a parameter θ are drawn from the conditional posterior distribution for θ, which can be calculated by multiplying by the likelihood and the prior that are both calculated at fixed values of all parameters except θ. In what follows, we will simply write $p(Y_p|\theta, \mathbf{x})$ as a shorthand for the likelihood in which all other parameters except θ, are fixed, and will assume the joint posterior to be multiplicatively separable. The sequential updating scheme depends on using conjugate priors in order that the posterior will have the same form as the prior, Fig. 3, so we parametrically characterize the marginal posteriors of each parameter using the distribution class of its conjugate prior, which in practice is often a good assumption. Not all parameters have likelihoods with a conjugate prior due to the presence of the exponential term, and for these, we will use the conjugate prior of the other term, under the assumption that the exponential term does not alter the shape of the distribution very much.

Sampling cluster intensity $\lambda_{0,c}$

Treating all other parameters besides $\lambda_{0,c}$ as constants, the likelihood Eq. 6 is $P(\Delta\mathbf{N}|\lambda_{0,c}, \mathbf{x}) \propto \lambda_{0,c}^{S_c} e^{-\lambda_{0,c}\beta}$, where $\beta = V_e(f_c, \sigma_c^2)$. The conjugate prior is the Gamma distribution Gamma$(\alpha_{\lambda_{0,c}}, \beta_{\lambda_{0,c}})$, where $\alpha_{\lambda_{0,c}}, \beta_{\lambda_{0,c}}$ are the posterior hyperparameters from the previous epoch. The conditional posterior from which we sample is then the Gamma distribution Gamma$(S_c + 1 + \alpha_{\lambda_{0,c}}, \beta + \beta_{\lambda_{0,c}})$, and the resulting posterior samples of $\lambda_{0,c}$, are well fit by a Gamma distribution, Fig. 3, middle column.

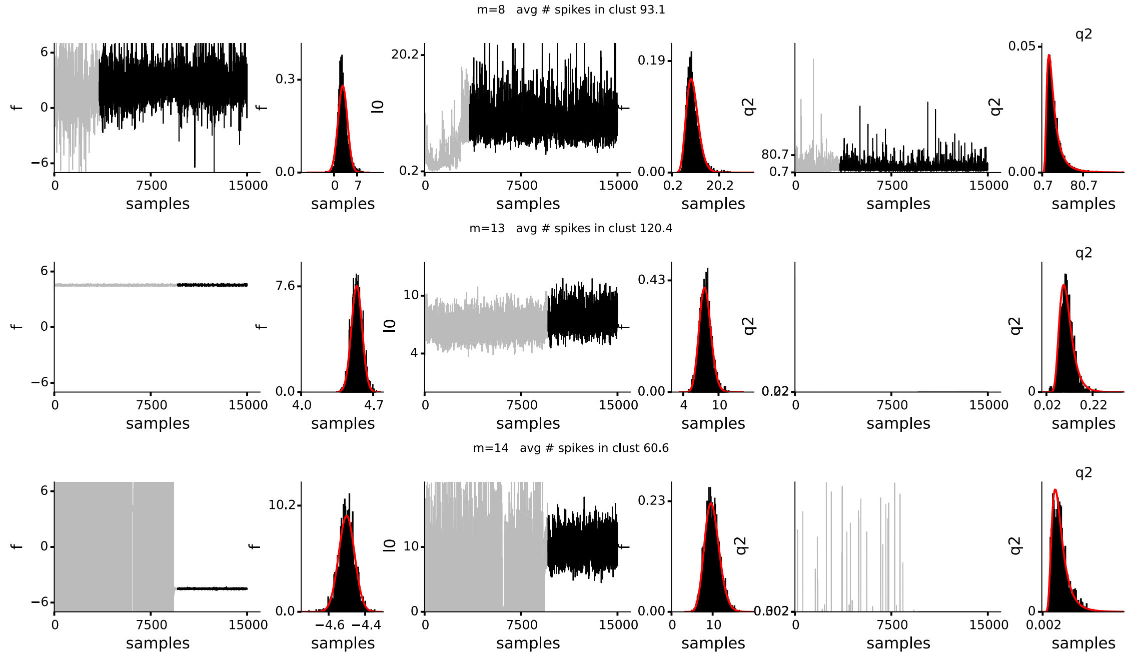

Figure 3: 3 selected chains of spatial and intensity parameters (f_c, λ_c, σ_c^2)from 3 typical clusters. The parameters are shown as samples from the chain from the start of the Gibbs procedure (grey), with samples deemed to be drawn from stationary distribution overlayed in black, starting at iteration S_c. Histograms of the stationary samples are shown to the right, with a fit by the model class of the conjugate prior (red). The assessment for stationary samples is done on a cluster by cluster basis.

Sampling cluster spatial center f_c

The likelihood Eq. 6 with terms involving f_c is

$$P(\Delta\mathbf{N}|f_c,\mathbf{x}) \propto \left\{ \exp\left[-\frac{\left(f_c - \frac{1}{S_c}\sum_{t\in\{t_s\}} \Delta N_t x_t\right)^2}{2(\sigma_c^2/S_c)} \right] \right\} \exp\left[-\lambda_{0,c} V_e(f_c,\sigma_c^2)\right], \tag{7}$$

where $\{t_s\} = \{t|\Delta N_t \geq 1\}$

There is no conjugate prior. However, we find that the sampled marginal posterior often is very close to a Gaussian, left column Fig. 3, which would become the prior to combine with the likelihood, so we use a Gaussian prior $N(\mu_{f_c}, \sigma_{f_c}^2)$ here, and obtain for the conditional posterior

$$P(f_c|\Delta\mathbf{N}, \mathbf{x}, \lambda_{0,c}, \sigma_c^2) \propto \exp\left[-\frac{(f_c - M_c)^2}{2\Sigma_s^2}\right] \exp\left[-\lambda_{0,c} V_e(f_c, \sigma_c^2)\right] \tag{8}$$

where

$$\bar{f}_c = \frac{1}{S_c}\sum_{t\in\{t_s\}} \Delta N_t x_t, \quad \bar{\sigma}_c^2 = \frac{\sigma_c^2}{S_c} \qquad M_c = \frac{\bar{f}_c \sigma_{f_c}^2 + \mu_{f_c}\bar{\sigma}_c^2}{\sigma_{f_c}^2 + \bar{\sigma}_c^2}, \quad \Sigma_c^2 = \frac{\sigma_{f_c}^2 \bar{\sigma}_c^2}{\sigma_{f_c}^2 + \bar{\sigma}_c^2} \tag{9}$$

Eq. 8 is not of a known distributional class, but f_c can be sampled from the numerically computed cumulative distribution function, which can be done efficiently by variable-resolution sampling at selected values of f_c, Fig. 4, and the samples fitted with a Gaussian. The variable-resolution sampling assumes the probability mass is concentrated, but not necessarily unimodal.

Sampling cluster spatial width σ_c^2

The conditional likelihood is

$$P(\Delta\mathbf{N}|\sigma_c^2, \mathbf{x}) \propto \left\{(2\pi\sigma_c^2)^{-\frac{S_c}{2}} \exp\left[-\frac{\sum_{t\in\{t_s\}}(f_c - x_t)^2}{2\sigma_c^2}\right]\right\} \exp\left[-\lambda_{0,c} V_e(f_c, \sigma_c^2)\right] \tag{10}$$

There is no conjugate prior. However, we find that the sampled marginal posterior often is very close to a inverse Gamma distribution, right column Fig. 3, so we use an inverse Gamma prior $\mathrm{IG}(\alpha_{\sigma_c^2}, \beta_{\sigma_c^2})$ here, and obtain for the conditional posterior

$$P(\sigma_c^2|\Delta\mathbf{N},\mathbf{x}) \propto \left\{(2\pi\sigma_c^2)^{-\alpha}\exp\left[-\frac{\beta}{\sigma_c^2}\right]\right\}\exp\left[-\lambda_{0,c}V_e(f_c,\sigma_c^2)\right] \tag{11}$$

where

$$\alpha \equiv \left(\alpha_{\sigma_c^2} + \frac{S_c}{2}\right) \qquad \beta \equiv \beta_{\sigma_c^2} + \sum_i^{S_c}\frac{(x_{s_i}-f_c)^2}{2} \tag{12}$$

Once again, Eq. 11 is not of known distributional class, but σ_c can be sampled using variable-resolution sampling as used for f_c. Here, the grid on which σ^2 is defined should itself be non-uniformly spaced, but rather scaled to be wider for larger values of σ^2, and variable-resolution applied to this non-uniformly spaced grid.

Sampling cluster mark center of $\boldsymbol{\mu}_c$

The conditional likelihood is

$$p(\Delta\mathbf{N},\mathbf{M}|\boldsymbol{\mu}_c) \propto N\left(\sum_{t\in\{t_s\}}\sum_{w}^{\Delta N_t}\mathbf{m}_{t,w}, \frac{1}{S_C}\boldsymbol{\Sigma}_c\right), \tag{13}$$

and the conjugate prior is $N(\boldsymbol{\mu}_{\boldsymbol{\mu}_c}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}_c})$. The conditional posterior is

$$p(\boldsymbol{\mu}_c|\Delta\mathbf{N},\mathbf{M}) = N\left[\left(\boldsymbol{\Sigma}_{\boldsymbol{\mu}_c}^{-1}+S_c\boldsymbol{\Sigma}_c^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{\mu}_c}^{-1}\boldsymbol{\mu}_{\boldsymbol{\mu}_c}+\boldsymbol{\Sigma}_c^{-1}\sum_{t\in\{t_s\}}\mathbf{m}_{t,w}\right), \left(\boldsymbol{\Sigma}_{\boldsymbol{\mu}_c}^{-1}+S_c\boldsymbol{\Sigma}_c^{-1}\right)^{-1}\right] \tag{14}$$

We approximate the posterior to be a multivariate normal, and estimate the hyperparameters using the sample mean and covariance of the samples.

Sampling cluster mark width $\mathbf{\Sigma}_c$

Rewriting the Gaussian in Eq. 13 using the cyclic permutation of the trace highlights the form conditional likelihood

$$p(\Delta \mathbf{N}, \mathbf{M}|\mathbf{\Sigma}_c) \propto (2\pi\,|\mathbf{\Sigma}_c|)^{-\frac{S_c}{2}} \exp\left\{-\frac{1}{2}\left[\sum_{t\in\{t_s\}}(\mathbf{m}_{t,w}-\boldsymbol{\mu}_c)^T(\mathbf{m}_{t,w}-\boldsymbol{\mu}_c)\right]\mathbf{\Sigma}_c^{-1}\right\}, \quad (15)$$

has the form of an inverse Wishart distribution, and the conjugate prior is $\mathrm{IW}(\nu_{\mathbf{\Sigma}_c}, \Psi_{\mathbf{\Sigma}_c})$. The conditional posterior is

$$p(\mathbf{\Sigma}_c|\boldsymbol{\mu}) = (2\pi\,|\mathbf{\Sigma}_c|)^{-\frac{w+K+1}{2}} \exp\left[-\frac{1}{2}\mathbf{P}\mathbf{\Sigma}_c^{-1}\right], \quad (16)$$

where

$$w = S_c + \nu_{\mathbf{\Sigma}_c}, \qquad \mathbf{P} = \left[\Psi_{\mathbf{\Sigma}_c} + \sum_{t\in\{t_s\}}(\mathbf{m}_{t,w}-\boldsymbol{\mu}_c)^T(\mathbf{m}_{t,w}-\boldsymbol{\mu}_c)\right] \quad (17)$$

Estimating the hyperparameters of an inverse Wishart distribution using maximum likelihood is difficult, so just for $\mathbf{\Sigma}_c$, its hyperparameters will be the means of the hyperparameters used in the conditional posterior, Eq. 17.

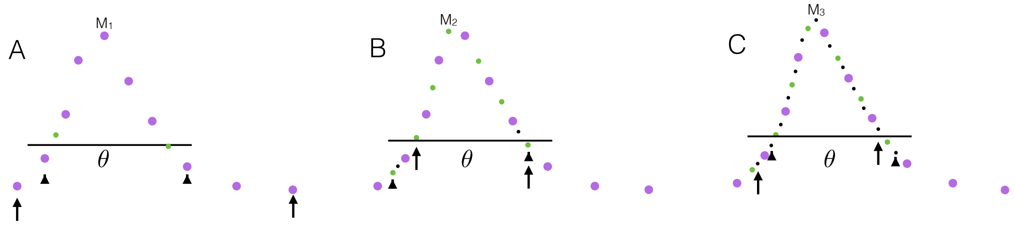

Figure 4: Variable-resolution sampling of conditional posterior. We first create a grid on which parameter θ is defined, and choose an integer $L > 0$, which determines how coarse the initial sampling on the first pass is done. A) As a first pass, the conditional posterior is coarsely sampled at every 2^Lth grid point, purple dots, over the range of the parameter θ, with $i' = \arg\max_i p(\theta_i|Y)$ and $M_1 = p(\theta_{i'}|Y)$. Starting at i' and for $r \ll 1$, search for a contiguous region of sampled values for is such that $p(\theta_i|Y) > rM_1$ (black triangles). This is the region where sampling at finer resolution should take place. B) Starting from left boundary of region found, shift by 2^{L-1} grid points to the left (red triangle), and sample at every 2^Lth grid points, until we hit the red triangle on the right. We once again find new $i' = \arg\max_i p(\theta_i|Y)$ and new limits. We repeat this process until we have a satisfactory number of sampled points within the high probability region. Now a sample can be drawn from the conditional posterior as the CDF can be easily calculated.

Updating number of clusters

Place fields can rapidly and dramatically change over the course of an experiment. Our approach here is not to infer the statistically optimal number of clusters to use at any point in the experiment, but ensure there is a high likelihood of observing the data, even at the expense of more than the optimal number of clusters. However, we would still like to be as concise as possible, so we use several heuristic criteria to assess whether to keep a cluster or not for the next epoch. The 1st criteria is that, on average, at least 1 data vector is assigned to the cluster throughout the Gibbs iterations. The 2nd is that the posterior distribution of the cluster center is reasonably well concentrated relative to the width of the cluster. We also require that at least 3000 iterations prior to the end of sampling be sufficiently stationary, to have some confidence that this cluster represents a well-isolated local optimum.

Updates of the encoding model

We suggested that the joint posterior to be approximately independent in its parameters. Because of this, we implement the sequential updates by using the marginal empirical posterior directly for each parameter as the prior for that parameter in the next epoch. However, if the signal is not stationary, but its statistical properties change relatively slowly in time, this will lead to an over-estimation of the variability. Absent predictions of how the place field and mark properties may change, this is equivalent to increasing the uncertainty of the posterior when we carry it over as a prior. For example, if Θ represented a place field center, and we believe that it could change by up to 10cm in 1 minute, no matter how certain we became of Θ at the nth time step, we could set a minimum width for $p(\Theta|Y^{n-1})$ that reflects a minimum of 10cm of uncertainty added every minute. For convenience, we denote the set of parameters that describe the cth mixture component by Θ_c. If the encoding step of two consecutive epochs are separated by time epoch T_e, at the end of each epoch, we modify the hyperparameters as follows:

$$\begin{aligned}
\alpha_{\lambda_{0,c}} &\rightarrow \alpha_{\lambda_{0,c}}\left(1-\tfrac{T_e}{\tau}\right) & \sigma^2_{f_c} &\rightarrow \sigma^2_{f_c}+T_e\sigma^2_{a,f} \\
\alpha_{\sigma^2_c} &\rightarrow \alpha_{\sigma^2_c}\left(1-\tfrac{T_e}{\tau}\right) & \Sigma^2_{\boldsymbol{\mu}_c} &\rightarrow \Sigma^2_{\boldsymbol{\mu}_c}+T_e\Sigma^2_{a,\boldsymbol{\mu}} \qquad (18)\\
\nu_{\Sigma_c} &\rightarrow \nu_{\Sigma_c}\left(1-\tfrac{T_e}{\tau}\right) & &
\end{aligned}$$

$\alpha_{\lambda_{0,c}}$, $\alpha_{\sigma^2_c}$, ν_{Σ_c}, $\sigma^2_{f_c}$ and $\Sigma^2_{\boldsymbol{\mu}_c}$ control the width of their respective prior distributions. τ controls the timescale in which adaptation occurs, and the $\sigma^2_{a,f}$ and $\Sigma_{a,\boldsymbol{\mu}}$ the amount of spread in the uncertainty of the width of the place field and marks. For the normal distributions, the uncertainty and the centers can be adjusted independently, but for the other parameters, modifying the above hyperparameters must be done in pairs, so that the mean or mode remains unchanged.

Simulated place fields and task

Simulations of hippocampal activity of a rat moving at variable speeds on a circular track were performed. Several neurons were defined, and their respective wave form features (marks) were generated from multivariate normal distributions whose parameters are specific to each neuron. Each neuron was then assigned one or more Gaussian place fields with accompanying intensity. Epochs defining when new data is used to update the model were either uniformly or randomly generated. We also defined low-amplitude spikes with broad tuning, and a cut-off threshold to mimic neural data used for clusterless decoding [19, 33, 20].

Hippocampal recordings and task

Extracellular data from 12 tetrodes in CA1 and CA2 of rat hippocampus was recorded while the rat traversed a W-shaped maze. Spikes were detected at a lower threshold than for preprocessing for traditional spike sorting [7, 34], as our goal is not necessarily clean isolation of individual units, but extraction of as much information as possible from the data [35], and all above-threshold spikes were used for decoding, even if they may be a mixture of noise and spiking from sources far from the tetrode [19, 33, 20]. However, lowering the threshold at which spikes are detected leads to rapid increase in the number of spikes. This has several undesirable consequences. The stochastic allocation into clusters requires calculating the

Mahalanobis distance from every data point to every cluster, so the number of computations required grows like $M \times N$, linearly with N, the number of spikes identified and used for fitting. With the larger N, deviations of the cluster shape from Gaussian become easier to detect, and our fitting procedure may assign multiple Gaussian components to the hash cluster. Because these ”spikes” may not even be of neuronal origin, investing too much time capturing the shapes of low-amplitude clusters may provide very little additional information about the physical position. However, past work suggests that these low-amplitude clusters are somewhat informative [20], so removing them entirely is undersirable, and requires us to set an arbitrary threshold for the hash, which might be difficult to do in a real-time setting. As such, we may also treat the threshold detailed in [34] as stochastic, and allow randomly allow some spikes lower than the threshold be allowed into our data. There tends to be more lower-amplitude spikes than high amplitude spikes, and this thinning allows us to keep some of these which might contain some information, without overwhelming the analysis with too many spikes that do not contribute as much information.

Rats moved through the maze in bouts of movement, interspersed with rest. The content of the hippocampal activity during movement is markedly different than during rest, with awake hippocampal replay events present in the latter [36, 37]. Because the spiking during rest may reflect not only reflect the current physical location, but internal states not accessible through monitoring of behavior, we built an encoding model using hippocampal activity and spatial location only when the animal was moving. Concatenating the periods of movement, we artificially constructed neural data of a rat in perpetual motion to test our decoder. epochs defining when new data is used to update the model were randomly generated.

The W-shaped maze is a 2-dimensional space, and the trajectory of the animal was linearized according to a scheme suggested by Deng (unpublished), so that a path going from home well H to the choice point (C), to the left (right) arm L (R) and back through C to home well H, was mapped to numerical position 0 to 1(-1) to 3(-3) to 5(-5) to 6(-6)(1, 5) and (2, 4). This mapping assigns different numbers to the same location depending on direction of movement, and requires some future information when mapping movement of animal as it leaves H, because whether the coordinates are negative or positive requires knowledge of whether it will eventually turn left or right.

Assessment of the encoding models

We assess JOYFULMoGs by using them to decode subsequent data. We decode neural activity using the marked point process filtering algorithm described in Deng *et al* [20], and assess the decode utilizing 2 metrics, the root mean squared error, calculated by taking the mean of the squared difference of between the *maximum a posterior* of the inferred filter position and the actual trajectory, and the time under 95% posterior, t_{95}, which is the time the actual trajectory is within the highest 95% posterior density regions.

Results

Sequentially accumulating evidence

JOYFULMoGs needs to solve 2 practical issues in real-time encoding and decoding. The first is the sequential estimation of a model as new data arrives, and the second is to adapt the model in cases where the underlying structure in the data changes throughout the experiment.

We assess JOYFULMoGs for these 2 criteria using simulations of a rat with a single tetrode implanted in the hippocampus navigating a circular track.

In the first set of simulations, the generative models for the place fields are static, and 12 epochs all of equal duration. 8 realizations for a recording scenario with 7 neurons being recorded are generated, Fig. 5. Encoding models for each realization are built in 2 ways: first where model parameters are estimated using only data from the most recent epoch (DSCRD), discarding all past information about the model, and second where sequential updating is performed (SQUP), and the past information about the model is carried over as a prior as described in Methods. The decode performance is compared for both approaches, and also with the performance of a model using ground truth parameters (GT) the data was generated with. The DSCRD strategy produces relatively constant decode errors over successive epochs, while the SQUP strategy approaches the performance of the GT after several epochs, and outperforms DSCRD, which is not surprising as it benefits from effectively a larger amount of training data, Fig. 5. Fig. 5A-C) right, show that for stationary data, whether a model is built using all the data at once or sequentially, the resulting model is essentially the same, as expected.

We note that the decode accuracy is a combination of how well our encoding model describes the JMIF, as well as how completely the place fields tile the space and how well the movement model describes actual movement. We have purposely simulated the rat's movement to alternate between randomly occurring bouts of slow and fast walking to create regions of high spiking that are due to high occupation (slow walking) rather than place field firing, to check that our method still models the place fields correctly. Our movement model however only captures the average movement speed, and is not well matched to the alternating speeds, leading to relatively large decoding errors.

Tracking changes in neural representation

Both biological changes, such as rapid changes in the neural representation [38, 39] and experimental changes, such as tetrode drift, may occur during the experiment. In the second simulation, we simulate place field and electrode drift, as well as more rapid changes over short timescales, such as new place fields appearing or disappearing over the course of the simulation. We will be comparing the SQUP to using a stale SQUP from several epochs previous. The stale encoding models should decode poorly near epochs where the neural representation is rapidly changing.

Fig. 6B shows the SQUP tracking the changing structure of the JMIF, as new place fields shift, appear or disappear, subjectively matching the changing features as compared to GT. Fig. 7A shows that SQUP decoder is able to keep up with rapid changes in the neural representation, and shows little difference in decode compared to the GT decoder, while the 7 epoch-old SQUP decoder shows rapid deterioration in decode quality as soon as there is rapid change in neural activity around epoch 10, Fig. 7B. We also see a gradual improvement in decode performance towards the later epochs, Fig. 7B and C, in agreement with Fig. 5, indicating that our method is able to accumulate evidence by sequential updates, while also keeping track of rapid changes present in the data.

Decoding physical movements

We sequentially decode the movement of a rat in the W-track on 2 different days, one an earlier and the other a later exposure to the track, Fig. 8. The trajectory through the track is visibly different for these 2 days, with the early exposure being more variable, and sometimes deviating greatly from the sequence the arms that must be visited for a reward. For example, in Fig. 8B on the left, a correct sequence should see the animal moving through the landmarks as H-C-L-C-H-C-R-C-H-C-L-C-H..., but this is violated around times $t = 150, 195, 310, 320, 390$. Fig. 8B on the right shows the sequence being visited correctly. A cursory examination of the mark-space raster plots for the early exposure suggest that on several tetrodes, neural representation appears to be changing, while for the late exposure, noticeable changes in the neural representation are not obvious looking at the mark-space raster plots. We show that JOYOUSMoGs indeed captures the changes in neural representation when present, and sequential updates to the model leads to significant improvements in decoding.

In the early exposure, our method tracked newly appearing place fields and possibly new neurons. The raster plot in Fig. 8 shows the activity visibly changing around time $t = 270$, and the epoch 1 and 4 JMIFs shown in Fig. 8D shows JOYFULMoGs capturing the newly appeared clusters at later epoch. On a later date, the data from the tetrodes look relatively stationary in both waveform marks and place field structure throughout the experiment, Fig. 8C right.

Example decoded trajectories are shown in Fig. 8E for the early and late exposures. In each day, an example decode using only 1 of the 15 tetrodes is shown above the decode when all 15 tetrodes are combined. Not surprisingly, combining tetrodes improves decode performance. We note that decodes shown are obtained by folding the map and the obtained densities, collapsing the the left and right distinction and the movement direction, so that the coordinates are now just H-C-LR, and represent only the linear displacement from home well. We do this because in the discontinuous scheme we used to build the encoding model, decode metrics like RMSE make little sense.

Fig. 9 shows a comparison between decodes in the early and late exposures. In each row of Fig. 9A), a comparison is made between using 1 tetrode versus combining all tetrodes, with a summary of the spread for 15 individual single tetrode decodes being represented by the error bars. We can see that in the early exposures, decode performance of the single tetrodes increases as more data is accumulated, while in the late exposure, performance remains relatively constant. This effect disappears when combining the tetrodes in this example, suggesting that perhaps only a few tetrodes are becoming more informative later on in the experiment. Fig. 9B) shows a comparison between using an up-to-date SQUP (age of EM=1) versus using a stale SQUP (age of EM=7). For the early exposure, there is a noticeable difference improvement in decoding when the up-to-date decoder is used, while in the late exposure, this difference is less pronounced. These results support the notion that neural representation is less stable earlier in an experience, and that neural representation can also change in relatively short timescales of seconds to minutes.

Discussion

We presented a Bayesian framework for a population model of neural spiking, JOYFULMoGs, that models the joint firing intensity dependent on waveform features and position, bypassing the need for spike sorting and fitting of place fields for each sorted neuron. JOYFULMoGs allows sequential updates to the model so it can incorporate the most recently observed data into the previous estimate of the model parameters, which also allows the model to track changes in the spatial and mark structure of the data, including appearances and disappearances of neurons and place fields. We demonstrated the method on simulated experimental data, showing increasingly accurate parameter estimation in later encoding epochs whether the data is stationary or dynamically changing. We also analyzed recordings from tetrodes placed in the CA3 region of hippocampus of a rat traversing a W-shaped environment, performing a continuous alternation task, and compared performance of early and late periods in training. We observed clear evidence of rapidly changing neural representation in early periods, and our model was clearly able to track and keep the population encoding model up-to-date. In addition, recent work on goodness of fit tests for marked point process models has shown JOYFULMoGs to produce models of activity that fit the data well [Long et al, accepted]. There are, however several outstanding issues that we need to address in order for JOYFULMoGs to be useful in a real-time experimental setting.

First, JOYFULMoGs is still not real-time for most datasets. The most computationally intensive part of our method is the stochastic allocation of spikes into clusters. For data with a relatively small number of spikes, estimation of JOYFULMoGs parameters can be done in real- or nearly real-time on a laptop. However, with larger number of spikes and putative clusters, the calculation of Mahalanobis distances between all spikes and clusters becomes a major bottleneck. This calculation involves $M \times N$ calculations of a distance, which can easily be broken into M independent calculations. After all the distances are calculated and spikes assigned to clusters, cluster parameter estimation can all proceed independently. A GPU implementation of Gibbs sampling of Gaussian mixture models has demonstrated [40] over an order of magnitude increase in performance over standard CPU implementations as we have done, suggesting that JOYFULMoGs can be similarly accelerated. Another approach to increasing performance, might be moving to an approximate Bayesian approach like the variational Bayes method to estimate the posterior distributions [41]. This may also allow a more optimal number of clusters to be inferred, or allow us to forgo our batch mode approach to estimation in favor continuous updates may prove more efficient [28, 29].

Second, complicated maze geometries are difficult to represent in 1-dimension, as JOYFULMoGs as presented requires. We presented the results of decoding on a W-maze using a simple linearization scheme, which discontinuously mapped 2-dimensional coordinates onto a 1-dimensional line. This approach may not be satisfactory for more complicated mazes with more than 1 choice point. Further, mazes that combine open areas with narrow maze segments cannot be meaningfully represented in 1-dimension. JOYFULMoGs is easily extensible to accommodate 2-dimensional spatial representations without a substantial increase in computing cost. Decoding using the non-linearized 2-dimensional coordinates of the W-maze have been demonstrated using JOYFULMoGs[in preparation]. There, we assume the spatial covariances of clusters are diagonal so the x and y components decouple, allowing fitting of the directions one at a time. Because the maze leaves many spatial locations unexplored, it is possible that place fields extend outside the accessible maze locations. However, without observation of such locations, conclusions cannot be drawn about the full extent of the place fields. Therefore, we only concern ourselves with the visible portion of place fields by including dummy trajectory data outside the visible maze, but without any spiking activity. This limits the Gibbs sampler from sampling cluster parameters that would make place fields extend far outside the maze. These simple modifications should extend to more complicated mazes, and mazes that combine narrow segments with wide open areas, allowing real-time analysis of more realistic environments. Future work will explore the use of JOYFULMoGs in such scenarios.

Acknowledgments

This work was supported by grants from the NIH (MH105174, NS094288) and the Simons Foundation (542971). We also thank Xinyi Deng, Ali Yousefi and Long Tao for helpful discussions.

References

[1] E. Stark and M. Abeles. Predicting Movement from Multiunit Activity. *Journal of Neuroscience*, 27(31):8387–8394, 2007.

[2] Jose M. Carmena, Mikhail A. Lebedev, Roy E. Crist, Joseph E. O’Doherty, David M. Santucci, Dragan F. Dimitrov, Parag G. Patil, Craig S. Henriquez, and Miguel A L Nicolelis. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biology*, 1(2):193–208, 2003.

[3] George W Fraser, Steven M Chase, Andrew Whitford, and Andrew B Schwartz. Control of a braincomputer interface without spike sorting. *Journal of Neural Engineering*, 6(5):055004, 2009.

[4] Cynthia A. Chestek, Vikash Gilja, Paul Nuyujukian, Justin D. Foster, Joline M. Fan, Matthew T. Kaufman, Mark M. Churchland, Zuley Rivera-Alvidrez, John P. Cunningham, Stephen I. Ryu, and Krishna V. Shenoy. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of Neural Engineering*, 8(4), 2011.

[5] William Bishop, Cynthia C Chestek, Vikash Gilja, Paul Nuyujukian, Justin D Foster, Stephen I Ryu, Krishna V Shenoy, and Byron M Yu. Self-recalibrating classifiers for intracortical braincomputer interfaces. *Journal of Neural Engineering*, 11(2):026001, 2014.

[6] David Sussillo, Sergey D. Stavisky, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. Corrigendum: Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 8:14490, 2017.

[7] Hernan Gonzalo Rey, Carlos Pedreira, and Rodrigo Quian Quiroga. Past, present and future of spike sorting techniques. *Brain Research Bulletin*, 119:106–117, 2015.

[8] Robert E Kass and Valérie Ventura. A Spike-Train Probability Model. *Neural Computation*, 1720:1713–1720, 2001.

[9] Anne C Smith and Emery N Brown. Estimating a state-space model from point process observations. *Neural computation*, 15(5):965–91, may 2003.

[10] W. Truccolo. A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2004.

[11] Jonathan W. Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[12] L. Paninski, E. N. Brown, S. Iyengar, and R. E. Kass. Statistical Models of Spike Trains. 2009.

[13] Maryam M. Shanechi, Rollin C. Hu, Marissa Powers, Gregory W. Wornell, Emery N. Brown, and Ziv M. Williams. Neural population partitioning and a concurrent brain-machine interface for sequential motor function. *Nature Neuroscience*, 15(12):1715–1722, 2012.

[14] Sonja Grün. Unitary Events in Multiple Single-Neuron Spiking Activity : I . Detection and Significance. *Neural computation*, 14:43–80, 2001.

[15] Victor Solo. Likelihood Functions for Multivariate Point Processes with. *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 4245–4250, 2007.

[16] Hideaki Shimazaki, Shun-ichi Amari, Emery N Brown, and Sonja Grün. State-Space Analysis of Time-Varying Higher-Order Spike Correlation for Multiple Neural Spike Train Data. 8(3), 2012.

[17] D. J. Daley and D. Vere-Jones. An introduction to the theory of point processes. 2003.

[18] Demba Ba, Simona Temereanca, and Emery N. Brown. Algorithms for the analysis of ensemble neural spiking activity using simultaneous-event multivariate point-process models. *Frontiers in Computational Neuroscience*, 8(February):1–13, 2014.

[19] F. Kloosterman, S. P. Layton, Z. Chen, and M. A. Wilson. Bayesian decoding using unsorted spikes in the rat hippocampus. *Journal of Neurophysiology*, 111(1):217–227, 2014.

[20] Xinyi Deng, Daniel F. Liu, Kenneth Key, Loren M. Frank, and Uri T. Eden. Clusterless Decoding of Position from Multiunit Activity Using a Marked Point Process Filter. *Neural computation*, 27:1438–1460, 2015.

[21] Danaipat Sodkomkham, Davide Ciliberti, Matthew A. Wilson, Ken Ichi Fukui, Koichi Moriyama, Masayuki Numao, and Fabian Kloosterman. Kernel density compression for real-time Bayesian encoding/decoding of unsorted hippocampal spikes. *Knowledge-Based Systems*, 94:1–12, 2016.

[22] Xinyi Deng, Daniel F. Liu, Mattias P. Karlsson, Loren M. Frank, and Uri T. Eden. Rapid classification of hippocampal replay content for real-time applications. *Journal of Neurophysiology*, 116(5):2221–2235, 2016.

[23] Jason E. Chung, Jeremy F. Magland, Alex H. Barnett, Vanessa M. Tolosa, Angela C. Tooker, Kye Y. Lee, Kedar G. Shah, Sarah H. Felix, Loren M. Frank, and Leslie F. Greengard. A Fully Automated Approach to Spike Sorting. *Neuron*, 95(6):1381–1394.e6, 2017.

[24] Valérie Ventura. Spike Train Decoding Without Spike Sorting. *Neural Computation*, 20(4):923–963, 2008.

[25] Valérie Ventura. Automatic Spike Sorting Using Tuning Information. *Neural Computation*, 21(9):2466–2501, 2009.

[26] Gopal Santhanam, Michael D. Linderman, Vikash Gilja, Afsheen Afshar, Stephen I. Ryu, Teresa H. Meng, and Krishna V. Shenoy. HermesB: A continuous neural recording system for freely behaving primates. *IEEE Transactions on Biomedical Engineering*, 54(11):2037–2050, 2007.

[27] Aharon Bar-Hillel, Adam Spiro, and Eran Stark. Spike sorting: Bayesian clustering of non-stationary data. *Journal of neuroscience methods*, 157(2):303–16, 2006.

[28] Jan Gasthaus, Frank Wood, G Dilan, and Yee Whye Teh. Dependent Dirichlet Process Spike Sorting. *Neural Information Processing Systems*, pages 1–8, 2008.

[29] Ana Calabrese and Liam Paninski. Kalman filter mixture model for spike sorting of non-stationary data. *Journal of Neuroscience Methods*, 196(1):159–169, 2011.

[30] Peter Sykacek and Stephen Roberts. Adaptive Classification by Variational Kalman Filtering. *Advances in Neural Information Processing Systems*, 15(1):737–744, 2003.

[31] Xiaoli Li. Signal processing in neuroscience. *Signal Processing in Neuroscience*, pages 1–288, 2016.

[32] Jean Diebolt and Christian P Robert. Distributions of Finite Mixture Estimation through

Bayesian Sampling. *Journal of the Royal Statistics Society. Series B*, 56(2):363–375, 1994.

[33] Sonia Todorova, Patrick Sadtler, Aaron Batista, Steven Chase, and Valérie Ventura. To sort or not to sort: the impact of spike-sorting on neural decoding performance. *Journal of Neural Engineering*, 11(5):056005, 2014.

[34] Rodrigo Quian Quiroga. Unsupervised spike detection and sorting Problem : detect and separate spikes. *Neural Computation*, 16:1661–1687, 2004.

[35] Emily R Oby, Sagi Perel, Patrick T Sadtler, Douglas A Ruff, Jessica L Mischel, David F Montez, Marlene R Cohen, Aaron P Batista, and Steven M Chase. Extracellular voltage threshold settings can be tuned for optimal encoding of movement and stimulus parameters. *Journal of Neural Engineering*, 13(3):036009, 2016.

[36] Margaret F Carr, Shantanu P Jadhav, and Loren M Frank. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience*, 14(2):147–53, 2011.

[37] Annabelle C. Singer, Margaret F. Carr, Mattias P. Karlsson, and Loren M. Frank. Hippocampal SWR Activity Predicts Correct Decisions during the Initial Learning of an Alternation Task. *Neuron*, 77(6):1163–1173, 2013.

[38] Gabriela Czanner, Uri T Eden, Sylvia Wirth, Marianna Yanike, Wendy a Suzuki, and Emery N Brown. Analysis of between-trial and within-trial neural spiking dynamics. *Journal of neurophysiology*, 99(5):2672–2693, 2008.

[39] Uri T Eden, Loren M Frank, Riccardo Barbieri, Victor Solo, and Emery N Brown. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural computation*, 16(5):971–98, 2004.

[40] Marc A. Suchard, Quanli Wang, Cliburn Chan, Jacob Frelinger, Andrew Cron, and Mike West. Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 19(2):419–438, 2010.

[41] Takashi Takekawa, Yoshikazu Isomura, and Tomoki Fukai. Spike sorting of heterogeneous neuron types by multimodality-weighted PCA and explicit robust variational Bayes. *Frontiers in Neuroinformatics*, 6(March):1–13, 2012.

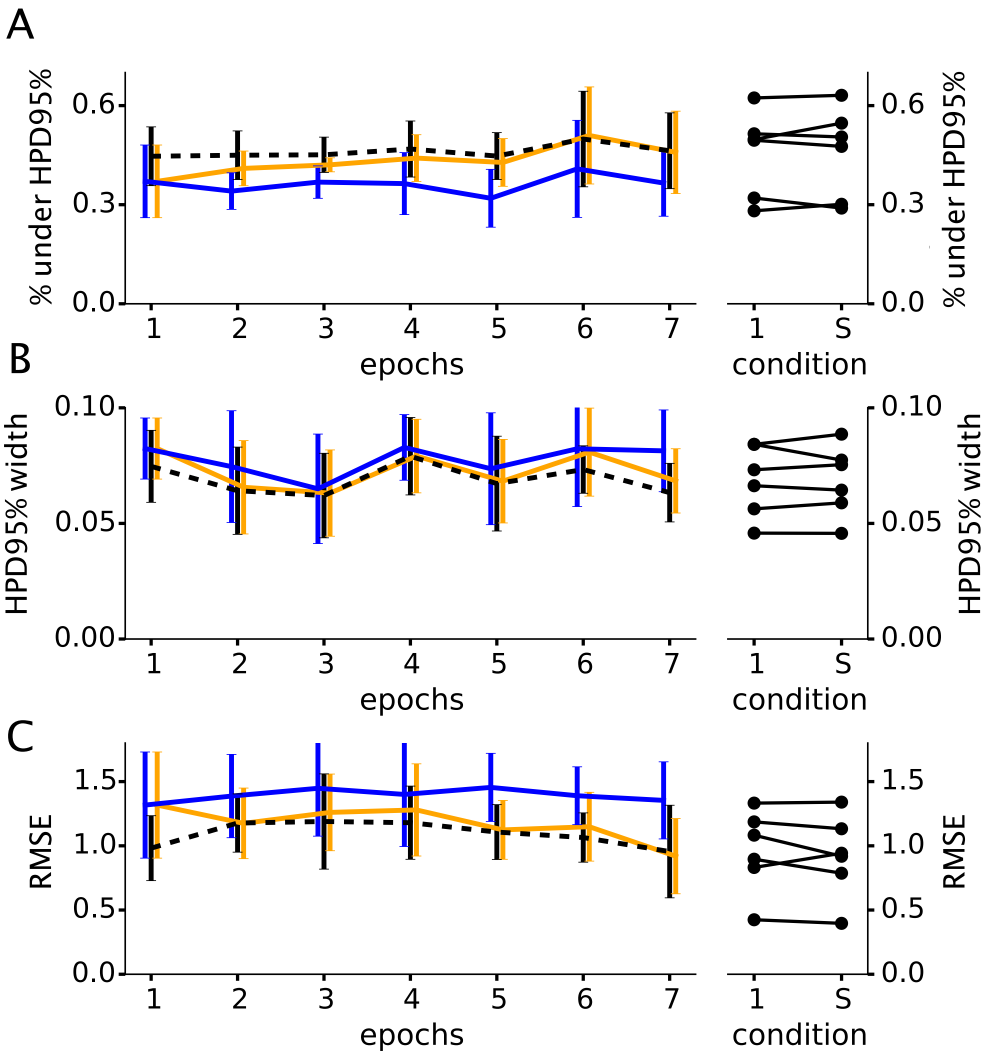

Figure 5: Assessment of sequentially-built encoding model on simulated, stationary data. For A-C), summary of performance metric of decode from 10 simulated experiments are plotted. Metrics are A) fraction of time actual path is under the 95% highest posterior density (HPD) area, B) the width of the 95% HPD area and C) the root mean-squared error (RMSE). Black, orange and blue lines show mean and standard deviation of decode metric for each epoch. Colors correspond to encoding model GT (ground truth parameter values at each epoch), SQUP (models built and updated sequentially) and DSCRD (no update, models built by discarding data more than 1 epoch old). Right panels show a comparison of mean decode performance of epoch 7 using model where data from all previous 6 epochs were used at once (condition 1), and the corresponding performance when model at epoch 7 was obtained by SQUP (condition S).

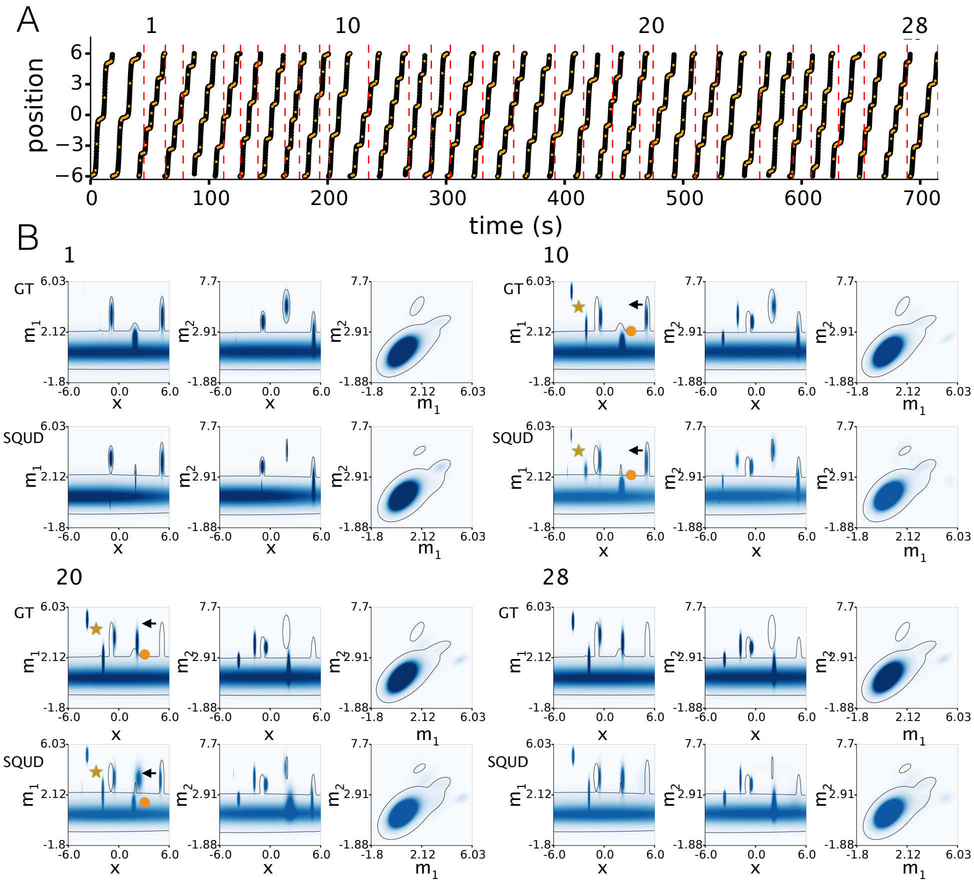

Figure 6: Encoding of simulated data with changing place fields. A) rat trajectory (black), overlayed with time and location of spikes (orange), with randomly chosen epochs shown by vertical, red dashed lines. The epoch numbers (1, 10, 20, 28) above trajectory correspond to epochs where JMIF is displayed in B). For each of those 4 sampled epochs, B) shows 2 rows of JMIF visualized by plotting 1 of 4 mark components against the position (left and middle), and against each other (right), for the GT (top) and SQUP (bottom), are shown. The widely-tuned component at the bottom represents hash spikes, which are low-amplitude spikes slightly above the discard threshold. The contour line shows the contour at 5% the maximum mark intensity of the first epoch, for comparison at the later epoch. 2 new place fields appear (near star), 1 place field disappears (near circle), and 1 place field rapidly shits (near arrow), around epoch 10.

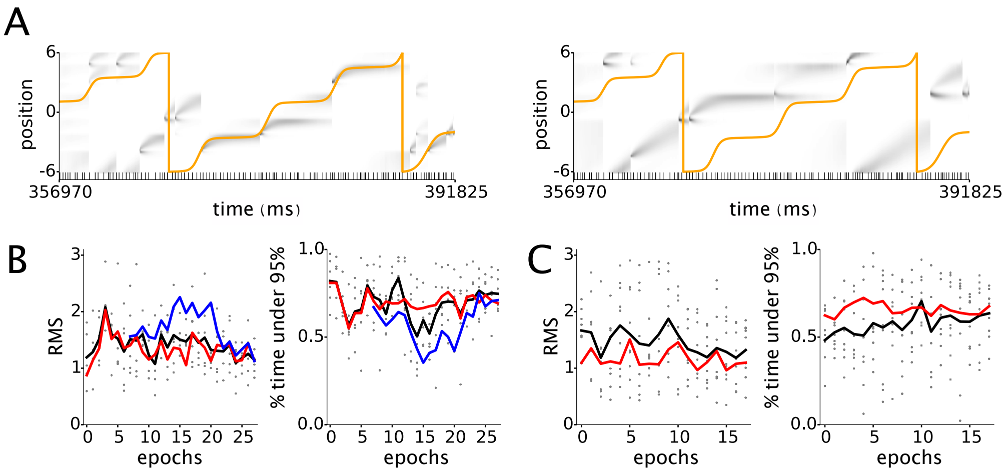

Figure 7: Decoding of simulated data with changing place fields. A) Decode of 1 epoch of sample realization, physical position in orange, posterior probability in grey. Decoding using SQUP (left), and stale SQUP (use model 7 epochs old). B) Summary of decodes of 9 realizations of simulated data with similar time course of changes as Fig. 6, with abrupt appearances and disappearances of place fields around epoch 10. Scatter plot (grey dots) of the RMS and % time under 95% HPD region for decoded trajectory using SQUP for each realization, while the black line is the average, the blue line the average using stale, 7 epoch-old SQUP, and red line is the average obtained using GT. C) Similar figure for another set of 9 realizations of a dataset where changes are more gradual, showing how the SQUP decode performance approaches decode performance using GT after many epochs.

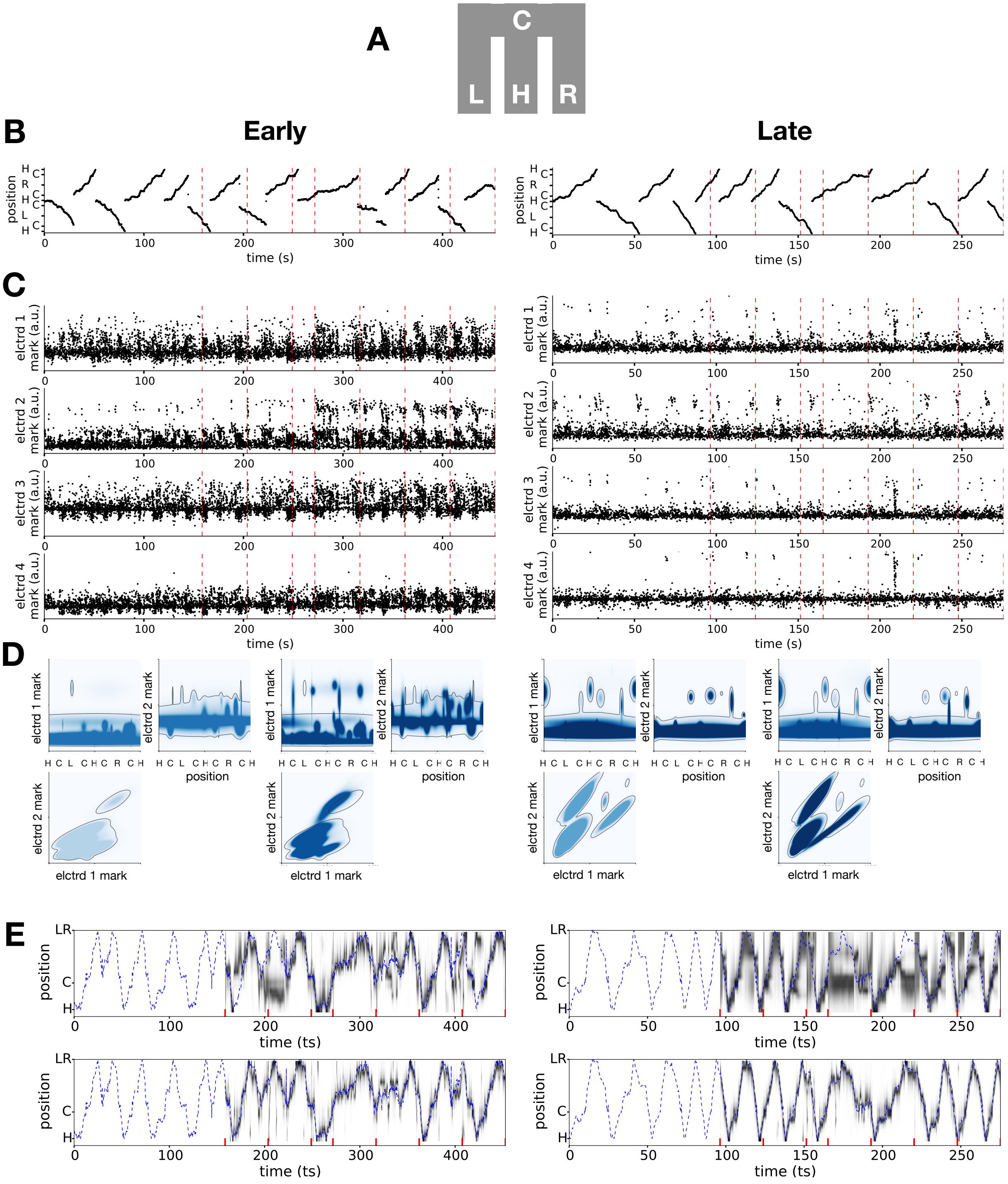

Figure 8: Decoding physical position of rat as it moves through a maze. A) Maze schematic, H: home well, C: choice point, L,R: reward wells on left and right. B) Rat trajectory in a W-maze, displayed in linearized 1-dimensional coordinates. Left and right show early and late exposure to the maze, respectively. Dashed red lines demarcate epochs used for sequential encoding. C) Marks recorded on each of the 4 channels of 1 example tetrode (out of 15) as a function of time. D) The marginalized mark intensity functions at epoch 1 and 5, shown using electrodes 1 and 2 and position. E) Decode example using sequentially updated encoding model. Top row is decoding result using the single tetrode shown above, and bottom row is decode combining all tetrodes. Decoded path displayed by collapsing left and right arms and heading direction to obtain linear displacement from home well.

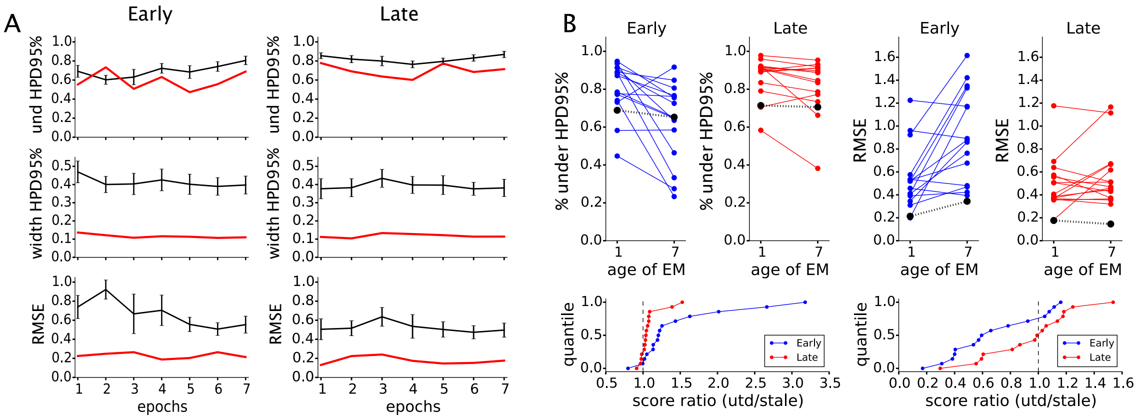

Figure 9: Quality of decode for the early and late exposures. A) From top, time within the 95% HPD region, the width of the HPD region and the RMSE of the physical location from the maximum posterior density for the early and late exposure. 0.1 in RMSE and HPD width corresponds to about 18 centimeters. Black lines are the mean values for the decodes done individually for the 15 tetrodes, with error bars representing 1/4 standard deviation, and the red line the results for decode combining likelihood from 15 tetrodes. B) Comparison of the decoding quality using each of the 15 tetrodes individually of the last epoch, roughly between 405-450 seconds in early exposure and 248-270 seconds in late exposure, using SQUP (age of EM=1), and using the encoding model obtained using data from only epoch 1 (age of EM=7), which is roughly the first 150 seconds in early exposure, and the first 100 seconds in late exposure. The black line is decoding quality using 15 tetrodes combined. C) ratios of the two sets of scores shown in B).