Appendices

A Controlled Sequential Monte Carlo

A key step in sampling both the cluster assignments and the cluster parameters of Algorithm 1 is computing the parameter likelihood $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ for an observation vector $\boldsymbol{y} = y_1, \ldots, y_T$ and a given set of parameters $\boldsymbol{\theta}$.

Recall the state-space model formulation.

$$\begin{aligned} x_1 \mid \boldsymbol{\theta} &\sim h(x_1; \boldsymbol{\theta}), \\ x_t \mid x_{t-1}, \boldsymbol{\theta} &\sim f(x_{t-1}, x_t; \boldsymbol{\theta}), && 1 < t \leq T, \\ y_t \mid x_t, \boldsymbol{\theta} &\sim g(x_t, y_t; \boldsymbol{\theta}), && 1 \leq t \leq T, \end{aligned}$$

A.1 Bootstrap Particle Filter

The bootstrap particle filter (BPF) of Doucet et al. (2001) is based on a sequential importance sampling procedure that iteratively approximates each filtering distribution $p(x_t \mid y_1, \ldots, y_t, \boldsymbol{\theta})$ with a set of S particles $\{x_t^1, \ldots, x_t^S\}$ so that

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} \left(\frac{1}{S} \sum_{s=1}^{S} g(x_t^s, y_t; \boldsymbol{\theta}) \right)$$

is an unbiased estimate of the parameter likelihood $p(\boldsymbol{y} \mid \boldsymbol{\theta})$. Algorithm 2 provides a review of this algorithm.

Algorithm 2 `BootstrapParticleFilter`($\boldsymbol{y}$, $\boldsymbol{\theta}$, f, g, h)

for $s = 1, \ldots, S$ **do**
  Sample $x_1^s \sim h(x_1)$.
  Weight $w_1^s = g(x_1^s, y_1; \boldsymbol{\theta})$.
end for
Normalize $\{w_1^s\}_{s=1}^S = \{w_1^s\}_{s=1}^S / \sum_{s=1}^S w_1^s$.
for $t = 2, \ldots, T$ **do**
  for $s = 1, \ldots, S$ **do**
    Resample ancestor index $a \sim \mathcal{C}\text{ategorical}(w_{t-1}^1, \ldots, w_{t-1}^S)$.
    Sample $x_t^s \sim f(x_{t-1}^a, x_t; \boldsymbol{\theta})$.
    Weight $w_t^s = g(x_t^s, y_t; \boldsymbol{\theta})$.
  end for
  Normalize $\{w_t^s\}_{s=1}^S = \{w_t^s\}_{s=1}^S / \sum_{s=1}^S w_t^s$.
end for
return Particles $\{\{x_1^s\}_{s=1}^S, \ldots, \{x_T^s\}_{s=1}^S\}$

A common problem with the BPF is that although its estimate of $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ is unbiased, this approximation may have high variance for certain observation vectors $\boldsymbol{y}$. The variance can be reduced at the price of increasing the number of particles, yet this often significantly increases computation time and is therefore unsatisfactory. To remedy our problem, we follow the work of Heng et al. (2017) in using controlled sequential Monte Carlo (cSMC) as an alternative to the standard bootstrap particle filter.

A.2 Twisted Sequential Monte Carlo

The basic idea of cSMC is to run several iterations of twisted sequential Monte Carlo, a process in which we redefine the model's state transition function f, initial prior h, and state-dependent likelihood g in a way that allows the BPF to produce lower-variance estimates without changing the parameter likelihood $p(\boldsymbol{y} \mid \boldsymbol{\theta})$. See also Guarniero et al. (2017) for a different iterative approach. Using a *policy* $\gamma = \{\gamma_1, \ldots, \gamma_T\}$ in which each γ_t is a

positive and bounded function, we define,

$$h^\gamma(x_1) := \frac{h(x_1)\cdot\gamma_1(x_1)}{H^\gamma},$$
$$f_t^\gamma(x_{t-1},x_t;\boldsymbol{\theta}) := \frac{f(x_{t-1},x_t;\boldsymbol{\theta})\cdot\gamma_t(x_t)}{F_t^\gamma(x_{t-1};\boldsymbol{\theta})}, \qquad 1<t\le T,$$

where $H^\gamma = \int h(x_1)\gamma_1(x_1)dx_1$ and $F_t^\gamma(x_{t-1};\boldsymbol{\theta}) = \int f(x_{t-1},x_t;\boldsymbol{\theta})\gamma_t(x_t)dx_t$ are normalization terms for the probability densities h^γ and f_t^γ, respectively. To ensure that the parameter likelihood estimate $\hat{p}(\boldsymbol{y}\mid\boldsymbol{\theta})$ remains unbiased under the twisted model, we define the twisted state-dependent likelihoods $g_1^\gamma,\dots,g_T^\gamma$ as functions that satisfy:

$$\hat{p}(\boldsymbol{x},\boldsymbol{y}\mid\boldsymbol{\theta}) = h^\gamma(x_1)\cdot\prod_{t=2}^T f_t^\gamma(x_{t-1},x_t;\boldsymbol{\theta})\cdot\prod_{t=1}^T g_t^\gamma(x_t,y_t;\boldsymbol{\theta})$$
$$h(x_1)\cdot\prod_{t=2}^T f(x_{t-1},x_t;\boldsymbol{\theta})\cdot\prod_{t=1}^T g(x_t,y_t;\boldsymbol{\theta}) = \frac{h(x_1)\gamma_1(x_1)}{H^\gamma}\cdot\prod_{t=2}^T\frac{f(x_{t-1},x_t;\boldsymbol{\theta})\gamma_t(x_t;\boldsymbol{\theta})}{F_t^\gamma(x_{t-1};\boldsymbol{\theta})}\cdot\prod_{t=1}^T g_t^\gamma(x_t,y_t;\boldsymbol{\theta})$$
$$\prod_{t=1}^T g(x_t,y_t;\boldsymbol{\theta}) = \frac{\gamma_1(x_1)}{H^\gamma}\cdot\prod_{t=2}^T\frac{\gamma_t(x_t)}{F_t^\gamma(x_{t-1};\boldsymbol{\theta})}\cdot\prod_{t=1}^T g_t^\gamma(x_t,y_t;\boldsymbol{\theta}).$$

This equality can be maintained if we define $g_1^\gamma,\dots,g_T^\gamma$ as follows,

$$g_1^\gamma(x_1,y_1;\boldsymbol{\theta}) := \frac{H^\gamma\cdot g(x_1,y_1;\boldsymbol{\theta})\cdot F_2^\gamma(x_1;\boldsymbol{\theta})}{\gamma_1(x_1)},$$
$$g_t^\gamma(x_t,y_t;\boldsymbol{\theta}) := \frac{g(x_t,y_t;\boldsymbol{\theta})\cdot F_{t+1}^\gamma(x_t;\boldsymbol{\theta})}{\gamma_t(x_t)}, \qquad 1<t<T,$$
$$g_T^\gamma(x_T,y_T;\boldsymbol{\theta}) := \frac{g(x_T,y_T;\boldsymbol{\theta})}{\gamma_T(x_T)}.$$

Thus, the parameter likelihood estimate of the twisted model is

$$\hat{p}^\gamma(\boldsymbol{y}\mid\boldsymbol{\theta}) = \prod_{t=1}^T\left(\frac{1}{S}\sum_{s=1}^S g_t^\gamma(x_t^s,y_t;\boldsymbol{\theta})\right).$$

The BPF is simply a degenerate case of twisted SMC in which $\gamma_t = 1$ for all t.

A.3 Determining the Optimal Policy γ^*

The variance of the estimate $\hat{p}^\gamma$ comes from the state-dependent likelihood g. Thus, to minimize the variance, we would like g_t^γ to be as uniform as possible with respect to x_t. Let the optimal policy be denoted γ^*. It follows that

$$\gamma_T^*(x_T) := g(x_T,y_T;\boldsymbol{\theta}),$$
$$\gamma_t^*(x_t) := g(x_t,y_t;\boldsymbol{\theta})\cdot F_{t+1}^{\gamma^*}(x_t;\boldsymbol{\theta}), \qquad 1\le t<T.$$

Under γ^*, the likelihood estimate $\hat{p}^{\gamma^*}(\boldsymbol{y}\,|\,\boldsymbol{\theta}) = H^{\gamma^*} = p(\boldsymbol{y}\,|\,\boldsymbol{\theta})$ has zero variance. However, it may be infeasible for us to use γ^* in many cases, because the BPF algorithm requires us to sample x_t from $f_t^{\gamma^*}$ for all t. For example, under γ^*, we would have

$$f_T^{\gamma^*}(x_{T-1},x_T;\boldsymbol{\theta}) \propto f(x_{T-1},x_T;\boldsymbol{\theta})\cdot\gamma_T^*(x_T) = f(x_{T-1},x_T;\boldsymbol{\theta})\cdot g(x_T,y_T;\boldsymbol{\theta}),$$

which may be impossible to directly sample from if f and g form an intractable posterior (e.g. if f is Gaussian and g is binomial). In such a case, we must choose a suboptimal policy γ.

A.4 Choosing a Policy γ for the Neuroscience Application

Recall the point-process state-space model, in which we have

$$\begin{aligned} h(x_1) &= \mathcal{N}(x_1 \mid x_0 + \mu, \psi_0), \\ f(x_{t-1}, x_t; \boldsymbol{\theta}) &= \mathcal{N}(x_t \mid x_{t-1} + \mu_t, \psi), \\ g(x_t, y_t) &= \text{Binomial}\left(R, \frac{\exp x_t}{1 + \exp x_t}\right), \end{aligned}$$

where we define the parameters $\boldsymbol{\theta} = \{\psi, \boldsymbol{\mu}\}$, where $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_T\}$.

Remark: In the main text, we set $\mu_t = 0$ for all $t \neq t_0$ and $\mu_{t_0} = \mu$. This makes the derivation below more general.

Here, we can show that $F_{t+1}^{\gamma^*}(x_t; \boldsymbol{\theta}) := \int f(x_t, x_{t+1}; \boldsymbol{\theta}) \gamma_{t+1}^*(x_{t+1}) dx_{t+1}$ must be log-concave in x_t. This further implies that for all t, $\gamma_t^*(x_t) := g(x_t, y_t) \cdot F_{t+1}^{\gamma^*}(x_t; \boldsymbol{\theta})$ is a log-concave function of x_t since the product of two log-concave functions is log-concave. Hence, we have shown that the optimal policy $\gamma^* = \{\gamma_1^*, \ldots, \gamma_T^*\}$ is a series of log-concave functions. This justifies the approximation of each $\gamma_t^*(x_t)$ with a Gaussian function

$$\gamma_t(x_t) = \exp(-a_t x_t^2 - b_t x_t - c_t), \quad (a_t, b_t, c_t) \in \mathbb{R}^3$$

and thus, $f_t^\gamma(x_{t-1}, x_t; \boldsymbol{\theta}) \propto f(x_{t-1}, x_t; \boldsymbol{\theta}) \cdot \gamma_t(x_t)$ is also a Gaussian density that is easy to sample from when running the BPF algorithm.

We want to find the values of (a_t, b_t, c_t) that enforce $\gamma_t \approx \gamma_t^*$ for all t. One simple way to accomplish this goal is to find the (a_t, b_t, c_t) that minimizes the least-squares difference between γ_t and γ_t^* in log-space. That is, given a set of samples $\{x_t^1, \ldots, x_t^S\}$ for the random variable x_t, we solve for

$$\begin{aligned} (a_t, b_t, c_t) &= \arg \min_{(a_t, b_t, c_t) \in \mathbb{R}^3} \sum_{s=1}^{S} \left[\log \gamma_t(x_t^s) - \log \gamma_t^*(x_t^s)\right]^2 \\ &= \arg \min_{(a_t, b_t, c_t) \in \mathbb{R}^3} \sum_{s=1}^{S} \left[-(a_t (x_t^s)^2 + b_t (x_t^s) + c_t) - \log \gamma_t^*(x_t^s)\right]^2 \end{aligned}$$

Also note that in a slight abuse of notation, we redefine for all $t < T$,

$$\gamma_t^*(x_t) := g(x_t, y_t) \cdot F_{t+1}^{\gamma}(x_t; \boldsymbol{\theta}_t)$$

because when performing approximate backwards recursion, it is not possible to analytically solve for the intractable integral $F_{t+1}^{\gamma^*}(x_t; \boldsymbol{\theta}_t)$.

In the aforementioned least-squares optimization problem, there is one additional constraint that we must take into account. Recall that $f_t^\gamma(x_{t-1}, x_t; \boldsymbol{\theta}) \propto f(x_{t-1}, x_t; \boldsymbol{\theta}) \cdot \gamma_t(x_t)$ is a Gaussian pdf that we sample from. Therefore, we must ensure that the variance of this distribution is positive, which places a constraint on γ_t and more specifically, the domain of (a_t, b_t, c_t). Using properties of Gaussians, we can perform algebraic manipulation to work out the following parameterizations of h^γ and f_t^γ:

$$\begin{aligned} h^\gamma(x_1) &= \mathcal{N}\left(x_1 \,\middle|\, \frac{\psi_0^{-1} \cdot (x_0 + \mu_1) - b_1}{\psi_0^{-1} + 2a_1}, \frac{1}{\psi_0^{-1} + 2a_1}\right) \\ f_t^\gamma(x_{t-1}, x_t; \boldsymbol{\theta}) &= \mathcal{N}\left(x_t \,\middle|\, \frac{\psi^{-1} \cdot (x_{t-1} + \mu_t) - b_t}{\psi^{-1} + 2a_t}, \frac{1}{\psi^{-1} + 2a_t}\right) \qquad \text{for all } t = 2, \ldots, T \end{aligned}$$

The corresponding normalizing terms for these densities are

$$\begin{aligned} H^\gamma &= \frac{1}{\sqrt{1 + 2a_1\psi_0}} \exp\left(\frac{\psi_0^{-1} \cdot (x_0 + \mu_1) - (b_1)^2}{2(\psi_0^{-1} + 2a_1)} - \frac{(x_0 + \mu_1)^2}{2\psi_0} - c_1\right) \\ F_t^\gamma(x_{t-1}; \boldsymbol{\theta}) &= \frac{1}{\sqrt{1 + 2a_t\psi}} \exp\left(\frac{\psi^{-1} \cdot (x_{t-1} + \mu_t) - (b_t)^2}{2(\psi^{-1} + 2a_t)} - \frac{(x_{t-1} + \mu_t)^2}{2\psi} - c_t\right) \qquad \text{for all } t = 2, \ldots, T \end{aligned}$$

Thus, to obtain (a_t, b_t, c_t) and consequently γ_t for all t, we solve the aforementioned least-squares minimization problem subject to the following constraints:

$$a_1 > -\frac{1}{2\psi_0} \qquad\qquad a_t > -\frac{1}{2\psi} \quad \text{for all } t = 2, \ldots, T$$

A.4.1 Full cSMC Algorithm

The full controlled sequential Monte Carlo algorithm iterates on twisted SMC for L iterations, building a series of policies $\gamma^{\langle 1 \rangle}, \gamma^{\langle 2 \rangle}, \ldots, \gamma^{\langle L \rangle}$ over time. Given two policies Γ' and γ, we can define

$$h^{\Gamma' \cdot \gamma}(x_1) \propto h^{\Gamma'}(x_1)\gamma_1(x_1) = h(x_1) \cdot \Gamma'_1(x_1) \cdot \gamma_1(x_1)$$
$$f_t^{\Gamma' \cdot \gamma}(x_{t-1}, x_t; \boldsymbol{\theta}) \propto f_t^{\Gamma'}(x_{t-1}, x_t; \boldsymbol{\theta}) \cdot \gamma_t(x_t) = f(x_{t-1}, x_t; \boldsymbol{\theta}) \cdot \Gamma'_t(x_t) \cdot \gamma_t(x_t)$$

We can see from these relationships that twisting the original model using Γ' and then twisting the new model using γ has the same effect as twisting the original model using a cumulative policy Γ where each $\Gamma_t(x_t) = \Gamma'_t(x_t) \cdot \gamma_t(x_t)$.

We state the full cSMC algorithm in Algorithm 3.

Algorithm 3 `ControlledSMC`($\boldsymbol{y}$, g, ψ, x_0, ψ_0, $\boldsymbol{\mu}$, L)

1: Define $f(x_{t-1}, x_t; \boldsymbol{\theta}) := \mathcal{N}(x_t \mid x_{t-1} + \mu_t, \psi)$ and $h(x_1) := \mathcal{N}(x_1 \mid x_0 + \mu_1, \psi_0)$.
2: Define parameters $\boldsymbol{\theta} = \{\psi, \boldsymbol{\mu}\}$.
3: Collect particles $\{x_1^s\}_{s=1}^S, \ldots, \{x_T^s\}_{s=1}^S$ from `BootstrapParticleFilter`($\boldsymbol{y}$, $\boldsymbol{\theta}$, f, g, h).
4: Initialize $\Gamma' = \{\Gamma'_1, \ldots, \Gamma'_T\}$ where $\Gamma'_t(x_t) = 1$ for all $t = 1, \ldots, T$.
5: Initialize $g_t^{\Gamma'}(x_t, y_t) = g(x_t, y_t)$ for all $t = 1, \ldots, T$.
6: Initialize $a_t^{\langle 0 \rangle} = 0, b_t^{\langle 0 \rangle} = 0, c_t^{\langle 0 \rangle} = 0$ for all $t = 1, \ldots, T$.

Algorithm 4 `ControlledSMC` (*continued*)

for $\ell = 1, \ldots, L$ **do**
  // Approximate backward recursion to determine policy and associated functions
  Define $\gamma_T^*(x_T) := g_T^{\Gamma'}(x_T, y_T)$.
  for $t = T, \ldots, 2$ **do**
    Solve $(a_t^{\langle\ell\rangle}, b_t^{\langle\ell\rangle}, c_t^{\langle\ell\rangle}) = \arg\min_{(a_t,b_t,c_t)} \sum_{s=1}^{S} \left[-(a_t(x_t^s)^2 + b_t(x_t^s) + c_t) - \log \gamma_t^*(x_t^s)\right]^2$ subject to $a_t > -1/(2\psi) - \sum_{\ell'=0}^{\ell-1} a_t^{\langle\ell'\rangle}$ using linear regression.
    Define new policy function $\gamma_t(x_t) := \exp(-a_t^{\langle\ell\rangle} x_t^2 - b_t^{\langle\ell\rangle} x_t - c_t^{\langle\ell\rangle})$.
    Define cumulative policy function $\Gamma_t(x_t) := \Gamma'_t(x_t) \cdot \gamma_t(x_t) = \exp(-A_t x_t^2 - B_t x_t - C_t)$ where $A_t := \sum_{\ell'=0}^{\ell} a_t^{\langle\ell'\rangle}$, $B_t := \sum_{\ell'=0}^{\ell} b_t^{\langle\ell'\rangle}$, and $C_t := \sum_{\ell'=0}^{\ell} c_t^{\langle\ell'\rangle}$.
    Define $f_t^{\Gamma}(x_{t-1}, x_t; \boldsymbol{\theta})$ and $F_t^{\Gamma}(x_{t-1}; \boldsymbol{\theta})$.
    if $t = T$ **then**
      Define $g_T^{\Gamma}(x_T, y_T) := \frac{g(x_T, y_T)}{\Gamma_T(x_T)}$.
    else
      Define $g_t^{\Gamma}(x_t, y_t) := \frac{g(x_t, y_t) \cdot F_{t+1}^{\Gamma}(x_t; \boldsymbol{\theta})}{\Gamma_t(x_t)}$.
    end if
    Define $\gamma_{t-1}^*(x_{t-1}) := g_{t-1}^{\Gamma'}(x_{t-1}, y_{t-1}) \cdot F_t^{\Gamma}(x_{t-1}; \boldsymbol{\theta}) / F_t^{\Gamma'}(x_{t-1}; \boldsymbol{\theta})$.
  end for
  Solve $(a_1^{\langle\ell\rangle}, b_1^{\langle\ell\rangle}, c_1^{\langle\ell\rangle}) = \arg\min_{(a_1,b_1,c_1)} \sum_{s=1}^{S} \left[-(a_1(x_1^s)^2 + b_1(x_1^s) + c_1) - \log \gamma_1^*(x_1^s)\right]^2$ subject to $a_1 > -1/(2\psi_0) - \sum_{\ell'=0}^{\ell-1} a_1^{\langle\ell'\rangle}$ using linear regression.
  Define new policy function $\gamma_1(x_1) := \exp(-a_1^{\langle\ell\rangle} x_1^2 - b_1^{\langle\ell\rangle} x_1 - c_1^{\langle\ell\rangle})$.
  Define cumulative policy function $\Gamma_1(x_1) := \Gamma'_1(x_1) \cdot \gamma_1(x_1) = \exp(-A_1 x_t^2 - B_1 x_t - C_1)$ where $A_1 := \sum_{\ell'=0}^{\ell} a_1^{\langle\ell'\rangle}$, $B_1 := \sum_{\ell'=0}^{\ell} b_1^{\langle\ell'\rangle}$, and $C_1 := \sum_{\ell'=0}^{\ell} c_1^{\langle\ell'\rangle}$.
  Define Γ-twisted initial prior $h^{\Gamma}(x_1)$ and H^{Γ}.
  Define $g_1^{\Gamma}(x_1, y_1) := \frac{H^{\Gamma} \cdot g(x_1, y_1) \cdot F_2^{\Gamma}(x_1; \boldsymbol{\theta})}{\Gamma_1(x_1)}$.
  // Forward bootstrap particle filter to sample particles and compute weights
  for $s = 1, \ldots, S$ **do**
    Sample $x_1^s \sim h^{\Gamma}(x_1)$.
    Weight $w_1^s = g_1^{\Gamma}(x_1^s, y_1)$.
  end for
  Normalize $\{w_1^s\}_{s=1}^S = \{w_1^s\}_{s=1}^S / \sum_{s=1}^S w_1^s$.
  for $t = 2, \ldots, T$ **do**
    for $s = 1, \ldots, S$ **do**
      Resample ancestor index $a \sim \mathcal{C}\text{ategorical}(w_{t-1}^1, \ldots, w_{t-1}^S)$.
      Sample $x_t^s \sim f_t^{\Gamma}(x_{t-1}^a, x_t; \boldsymbol{\theta})$.
      Weight $w_t^s = g_t^{\Gamma}(x_t^s, y_t)$.
    end for
    Normalize $\{w_t^s\}_{s=1}^S = \{w_t^s\}_{s=1}^S / \sum_{s=1}^S w_t^s$.
  end for
  Update $\Gamma' \leftarrow \Gamma$.
end for
return Likehood estimate $\hat{p}^{\Gamma}(\boldsymbol{y} \mid \boldsymbol{\theta})$.

B Clustering Cue Responses: Figures of Additional Clusters

Following cluster selection, we identified a total of 9 different types of responses to the cue. The overlaid rasters from two of these clusters are shown in Figure 4. Figures 8-14 shows the additional clusters. Together, Figure 4 and 8-14 demonstrate the ability of our approach to cluster neural responses to a stimulus into a set of clusters whose overlaid rasters are not unlike typical responses from single neurons.

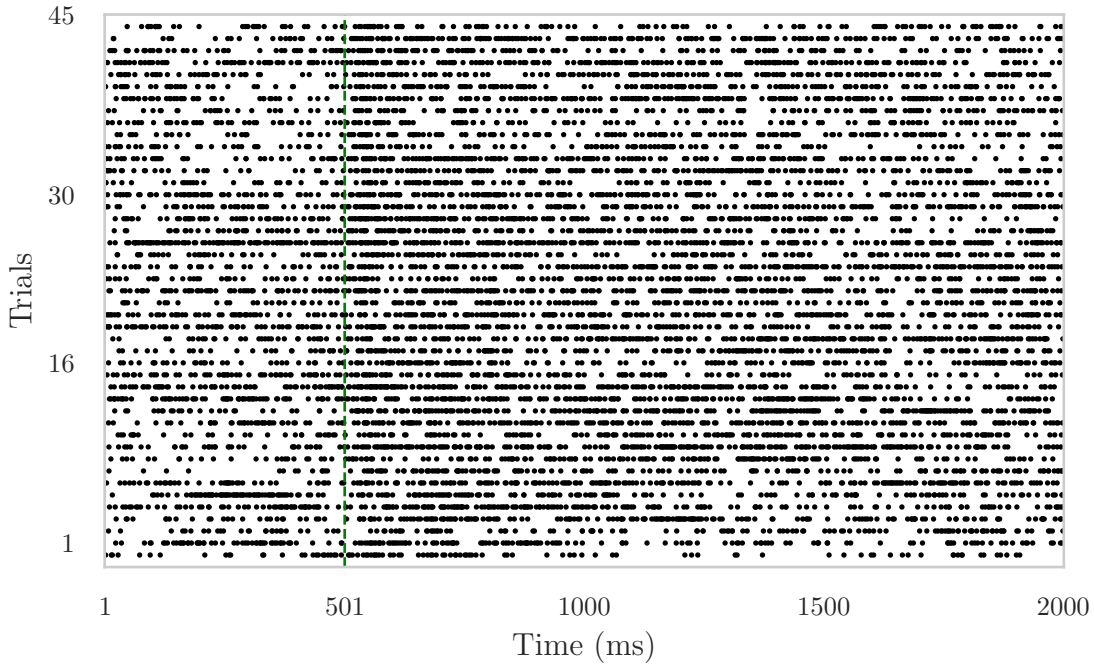

Figure 8: Overlaid raster plots of neurons with very excited and unsustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

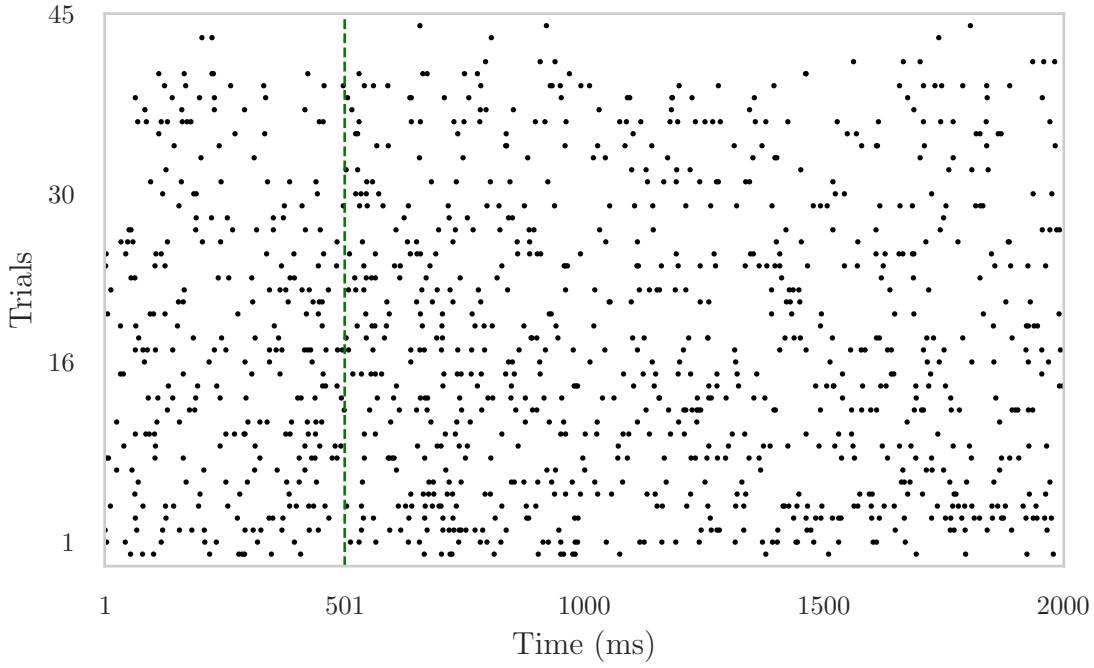

Figure 9: Overlaid raster plots of neurons with moderately excited and unsustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

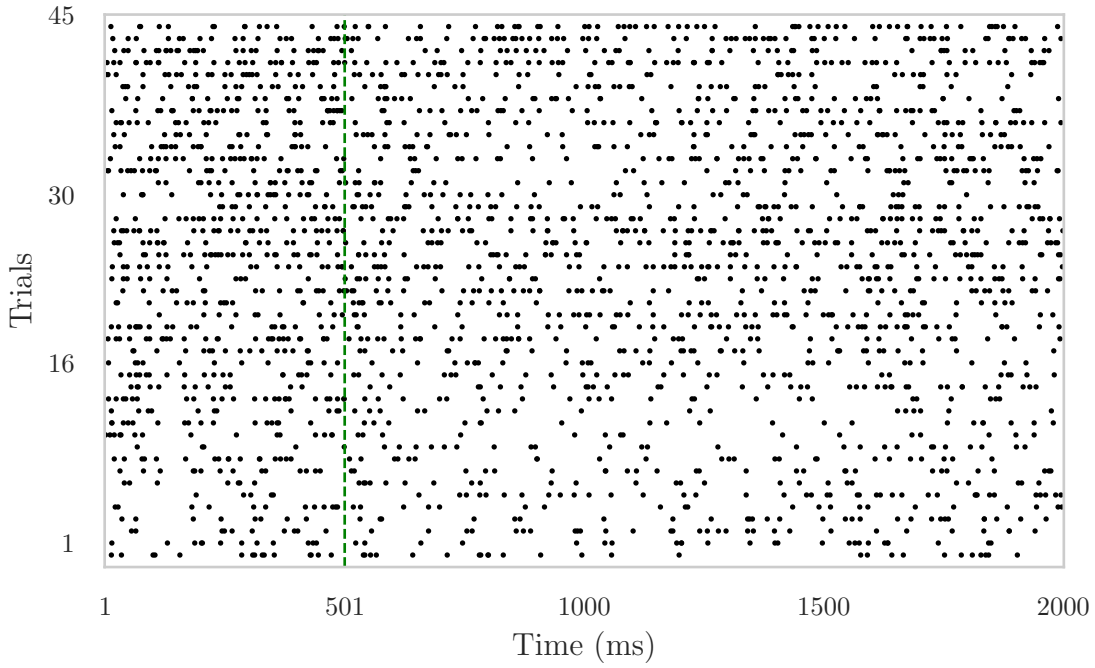

Figure 10: Overlaid raster plots of neurons with very inhibited and unsustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

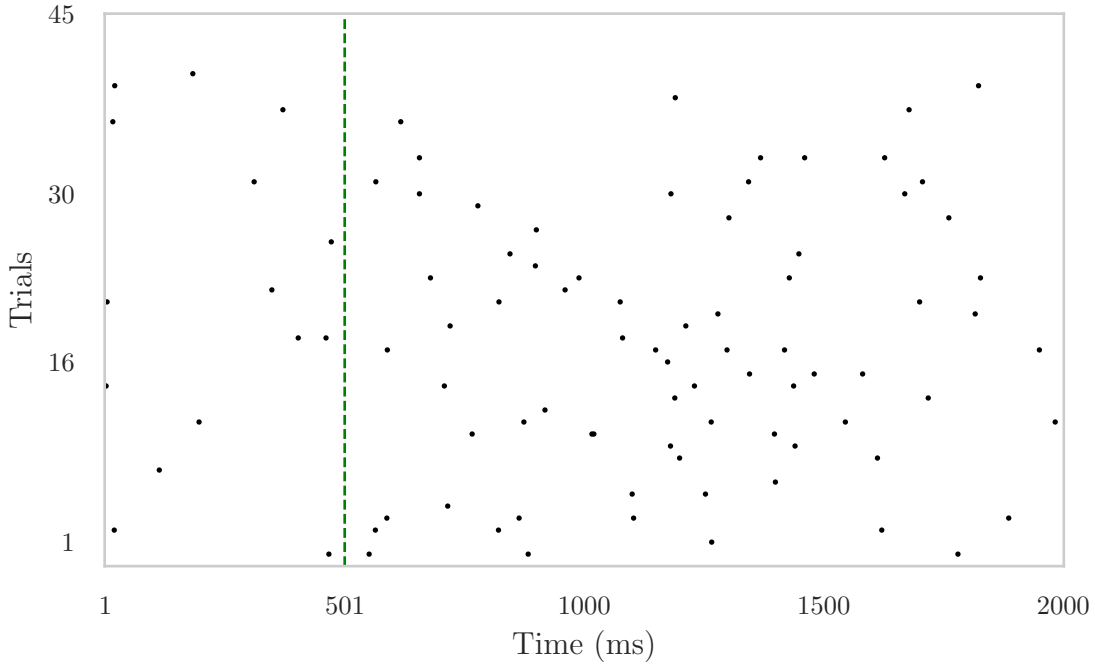

Figure 11: Overlaid raster plots of neurons with moderately excited and sustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

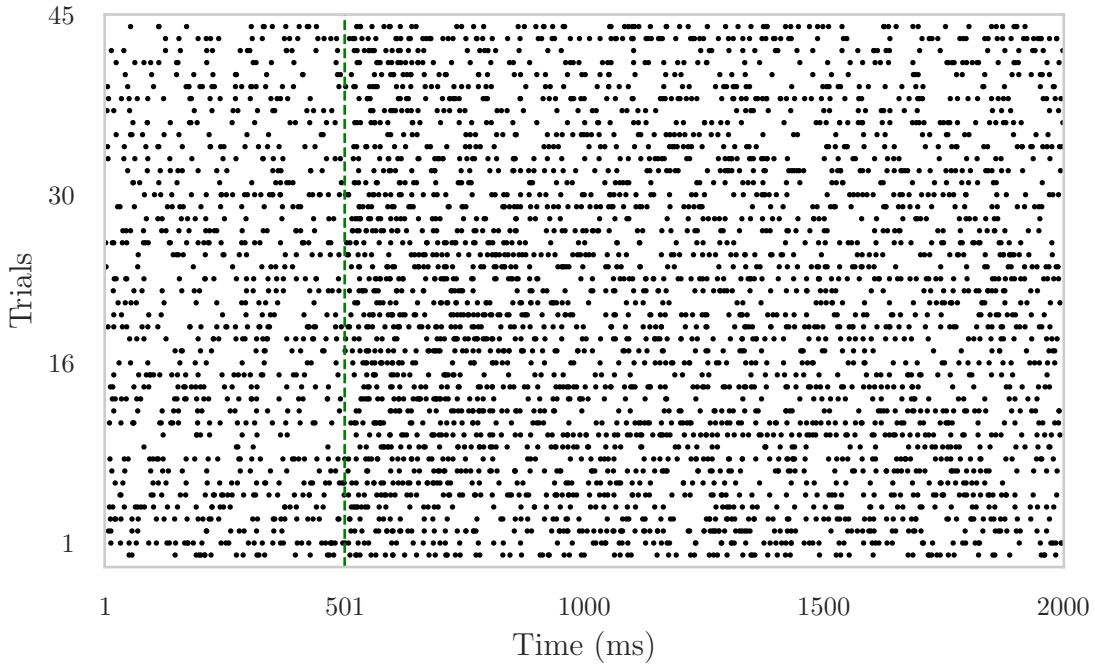

Figure 12: Overlaid raster plots of neurons with very excited and unsustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

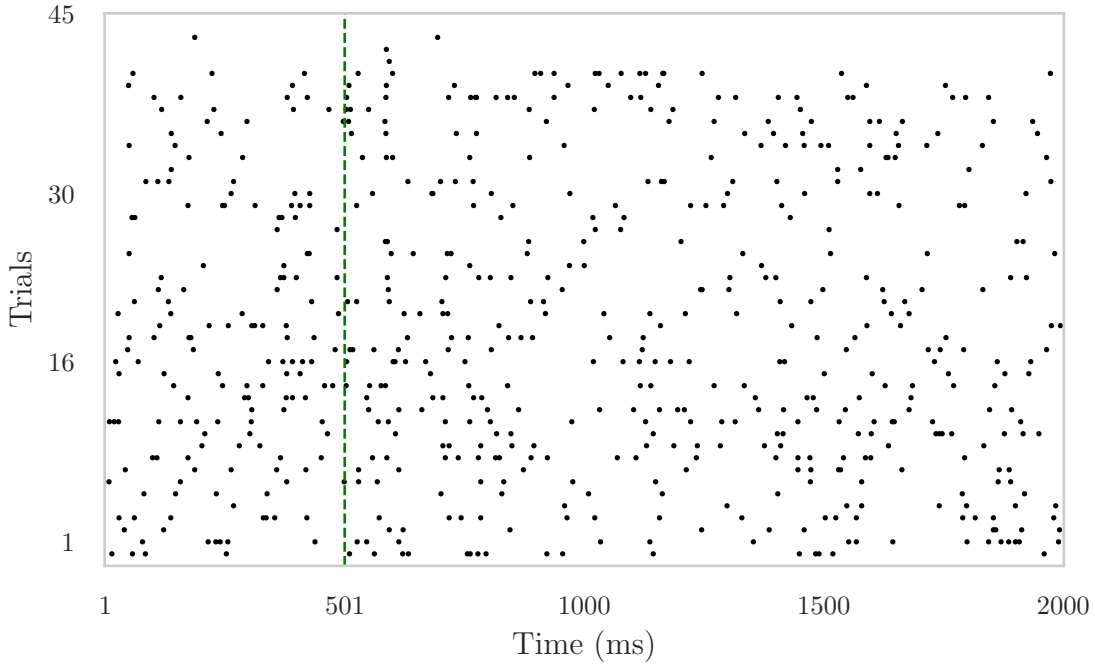

Figure 13: Overlaid raster plots of neurons with slightly inhibited and sustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

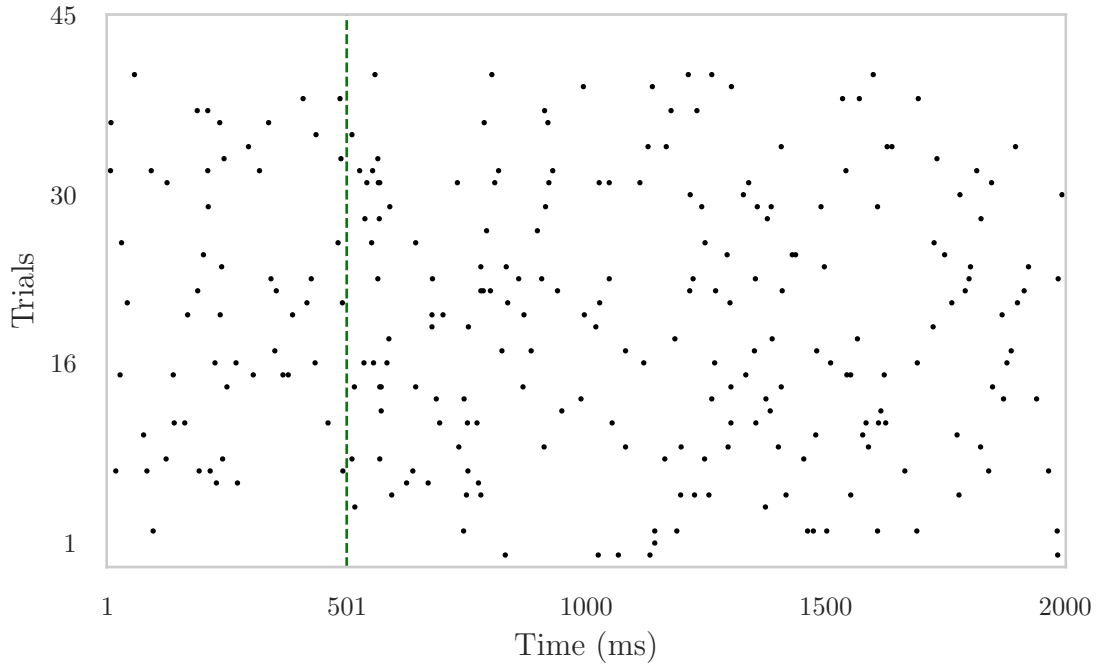

Figure 14: Overlaid raster plots of neurons with slightly excited and sustained responses. A black dot indicates a spike from at least one of the neurons in the corresponding cluster. The vertical green line indicates cue onset.

C Clustering Shock Responses: How subtle is the effect Figure 6(b) (main text)?

The fact that Figure 6(b) comprises neurons that are inhibited in response to the shock is not visually apparent. We pick two neurons from the cluster and demonstrate that the shock has an inhibitory, albeit subtle, effect on their response. In particular, we compare the empirical estimate of the rate at all trials $1, \cdots, 45$ to an estimate of the rate obtained from the DPnSSM model.

We compute the empirical rate of events at trial t in units of Hz as $\hat{\lambda}_t^{\text{emp}} = 1000 \cdot y_t^{(n)}/2000$, where the factor of 1000 is to convert the empirical probability estimates to units of Hz.

Suppose the cluster selection method described in the main text selects the samples from Gibbs iteration i^*. For a given neuron n with cluster assignment $z^{(n)\langle i^*\rangle}$ and parameter $\theta^{(z^{(n)\langle i^*\rangle})}$ obtained from hierarchical clustering applied to the co-occurence matrix (please see main text), we use cSMC to generate $S = 64$ samples $x_1^s, \ldots, x_T^s$, where for each $t = 1, \ldots, T$,

$$x_t^s \sim p(x_t^{(n)} | z^{(n)\langle i^*\rangle}, \theta^{(z^{(n)\langle i^*\rangle})}, y_1^{(n)}, \ldots, y_t^{(n)}) \tag{1}$$

for all s. We then compute

$$\hat{x}_t = \frac{1}{S} \sum_{s=1}^{S} x_t^s. \tag{2}$$

Finally, as the DPnSSM estimate of the rate in Hz, we use

$$\hat{\lambda}_t^{\text{dpnssm}} = 1000 \left(\frac{\exp \hat{x}_t}{1 + \exp \hat{x}_t} \right) \tag{3}$$

Figure 15 shows two neurons from the cluster in Figure 6(b). Overall, the empirical rate from each neuron indicates a downward trend, that is accentuated around trial 16, the trial when shock is delivered. In the DPnSSM, this is captured by the abrupt change in the empirical rate at trial 16, which indicates the fact that

these neurons are inhibited, albeit very subtly so, in response to the shock. In other words, despite the fact this effect is not obvious from the overlaid raster of Figure 6(b), Figure 15 indicates that the DPnSSM is able to identify a subtle effect that can be seen in the raw data.

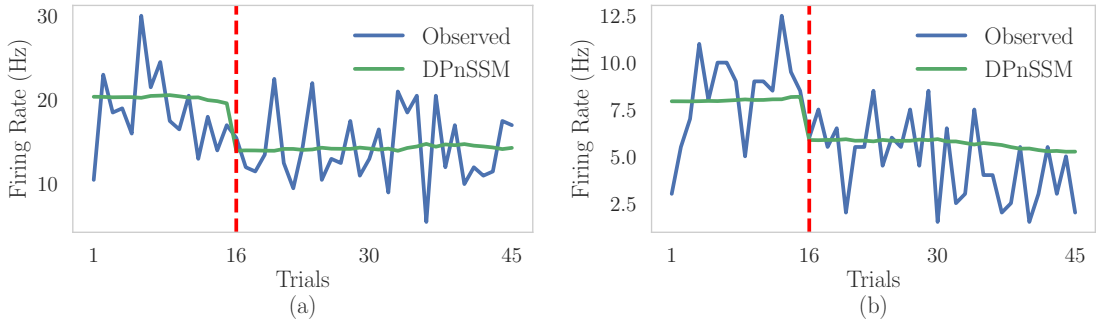

Figure 15: Two representative neurons from the cluster corresponding to Figure 6(b). The DPnSSM state sequence is able to track the overall trend in the observed data and correctly characterize the response to the shock at trial 16 as slightly inhibited.

References

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Guarniero, P., Johansen, A. M., and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647.

Heng, J., Bishop, A. N., Deligiannidis, G., and Doucet, A. (2017). Controlled sequential monte carlo. *arXiv preprint arXiv:1708.08396*.