S1 Appendix
Supplementary Methods for “Dynamic Estimation of Auditory Temporal Response Functions via State-Space Models with Gaussian Mixture Process Noise”

Sina Miran¹, Alessandro Presacco², Jonathan Z. Simon²,³,⁴, Michael C. Fu²,⁵, Steven I. Marcus²,³, Behtash Babadi²,³, *

¹ Starkey Hearing Technologies, Eden Prairie, MN, USA
² Institute for Systems Research, University of Maryland, College Park, MD, USA
³ Department of Electrical & Computer Engineering, University of Maryland, College Park, MD, USA
⁴ Department of Biology, University of Maryland, College Park, MD, USA
⁵ Robert H. Smith School of Business, University of Maryland, College Park, MD, USA

* behtash@umd.edu

In this appendix, we provide: (i) detailed derivations of the two approaches used for computing the expectations in Eqs. (9), (10), and (11), (ii) the criteria for model order selection, and (iii) details of the MSAR estimation procedure. In what follows, we use the notation $\mathbf{y}_{n_1:n_2}$ to denote the set of observations from $n_1$ to $n_2$, i.e., $\mathbf{y}_{n_1:n_2}$, and similarly define $\mathbf{x}_{n_1:n_2}$ and $\mathbf{z}_{i_1:i_2}$ for $\mathbf{x}_{n_1:n_2}$ and $\mathbf{z}_{i_1:i_2}$, respectively.

**Approach 1: Monte Carlo Approximations**

One way to approximate the expectations in the update equations of the M-step is to utilize Monte Carlo methods. Let $\mathbf{x}_{(i,0):(i,W)}^{(u)}$ for $u = 1, \ldots, U$ denote a total of $U$ sample paths, i.e., particles, with corresponding weights of $\omega_i^{(u)}$ inside the $i^{th}$ window to approximate the joint smoothing density of $\mathbf{X}_{(i,0):(i,W)}^{(u)} | \mathbf{Y}_i^N, \hat{\Theta}^{(l)}$ for $i = 1, \ldots, K$. Using this particle approximation, the update equations of the M-step become:
Algorithm S1. A Forward-Backward Particle Smoothing Alg.

Input: state-space model in (1) and parameter estimate $\hat{\Theta}^{(t)}$.

Output: sample paths $x_{(i,0):(i,W)}^{(u)}$ and their weights $\omega_{i,m}^{(u)}$.

1. Initialize $x_{0}^{(u)}$ and their filtering weights $\tilde{\omega}_{0}^{(u)} = 1/U$.
2. for $i = 1 : K$ do
   3. Sample $z_{i}^{(u)}$ according to $p_{1,M}^{(u)}$.
   4. Sample $x_{(i,0):(i,W)}^{(u)}$ using $x_{(i,0)}^{(u)}$ as the starting point and $z_{i}^{(u)}$ as the active Gaussian component.
   5. $\omega_{1}^{(u)} = \prod_{j=1}^{W} P\left( y_{(i,j)}^{(u)} \mid x_{(i,j)}^{(u)} \right)$.
   6. Normalize the weights such that $\sum_{u=1}^{U} \omega_{1}^{(u)} = 1$.
   7. Resample $x_{(i,W)}^{(u)}$ for next window according to $\tilde{\omega}_{1}^{(u)}$.
8. end for
9. Initialize the smoothing weights $\omega_{K}^{(u)} = \tilde{\omega}_{K}^{(u)}$.
10. for $i = K - 1 : 1$ do
   11. $\omega_{i}^{(u)} = \tilde{\omega}_{i}^{(u)} \sum_{u' = 1}^{U} \sum_{u'' = 1}^{U} P\left( x_{(i+1,1)}^{(u')} \mid x_{(i,W)}^{(u'')}, \hat{\Theta}^{(t)} \right) \omega_{i+1}^{(u'')}$.
12. end for

where $y_{(i,j)}^{(u)} = x_{(i,j)}^{(u)} - f_{(i,j)} \left( x_{(i,j-1)}^{(u)} \right)$, and $\tilde{\omega}_{i,m}^{(u)}$ is defined similarly to $\tilde{\omega}_{i,m}^{(u)}$ in (8) with $\tilde{\pi}_{(i,j),m}^{(t)}$’s evaluated at $x_{(i,0),(i,W)}^{(u)} = x_{(i,0):(i,W)}^{(u)}$ and $\Theta = \hat{\Theta}^{(t)}$ in Eq. (6). Particle smoothing approaches are SMC methods that provide the sample paths $x_{(i,0):(i,W)}^{(u)}$ and their respective weights $\omega_{i,m}^{(u)}$. The class of algorithms using SMC within EM for SSMs are referred to as SMCEM [2]. A forward-backward particle smoothing algorithm is presented in Algorithm S1 as an example of how the approximating particles can be computed.

**Remark 1.** In general, particle smoothing approaches are computationally intensive, especially for high-dimensional problems, which limits their application compared to

$$\hat{\rho}^{(t+1)} = \frac{1}{K} \sum_{i=1}^{K} \sum_{u=1}^{U} \omega_{i,m}^{(u)} \xi_{i,m}^{(u)} \rho_{i,m}^{(t+1)}, \quad (S1)$$

$$\hat{\mu}^{(t+1)} = \frac{\sum_{i=1}^{K} \sum_{u=1}^{U} \omega_{i,m}^{(u)} \xi_{i,m}^{(u)} \sum_{j=1}^{W} y_{(i,j)}^{(u)} - \hat{\mu}^{(t+1)}}{W \sum_{i=1}^{K} \sum_{u=1}^{U} \omega_{i,m}^{(u)} \xi_{i,m}^{(u)} \mu_{i,m}^{(t+1)}}, \quad (S2)$$

$$\Sigma^{(t+1)} = \frac{\sum_{i=1}^{K} \sum_{u=1}^{U} \omega_{i,m}^{(u)} \xi_{i,m}^{(u)} \sum_{j=1}^{W} \left( y_{(i,j)}^{(u)} - \hat{\mu}^{(t+1)} \right) \left( y_{(i,j)}^{(u)} - \hat{\mu}^{(t+1)} \right)^{T} }{W \sum_{i=1}^{K} \sum_{u=1}^{U} \omega_{i,m}^{(u)} \xi_{i,m}^{(u)} } \quad (S3)$$
particle filtering methods. In our setting, densities of dimension $d_x(W + 1)$ have to be approximated by particles. The forward-backward method in Algorithm S1 simply re-weights the filtering particles according to future observations and incurs an $O(U^2)$ cost. The two-filter particle smoother [3] samples the particles in the smoothing step but has a similar computational cost. In [3], an approximation based on spatial-index methods is introduced to reduce the computational cost to $O(U \log U)$. Finally, a particle smoothing method with $O(U)$ cost (similar to that of particle filtering) is developed in [4]. However, it operates under the assumption of minimal posterior dependence between $x_{n-1}$ and $x_{n+1}$ when sampling for the smoothing density of $x_n$.

**Approach 2: Closed-Form Approximations**

In this section, we consider a linear SSM, i.e., $f_n(x_{n-1}) = A_n x_{n-1}$ and $g_n(x_n) = C_n x_n$ in (1), to exploit the GM formulation of the smoothing densities [5]. Techniques such as the extended Kalman filter [6] or the unscented Kalman filter [7] are often used to approximate the general state-space model of Eq. (1) with a linear model. We introduce an approximation to the expectations in the M-step which allows to employ GM smoothing densities for computing the updated parameters in EM. This is akin to the application of EM in linear Gaussian SSMs [8]. Then, we construct an algorithm to efficiently compute the required smoothing densities in closed-form for our setting. As a result, the computational cost of the M-step would be comparable to performing parallel instances of fixed-interval smoothing, each corresponding to a component of the GM process noise.

We first consider a 0th-order Taylor expansion for $\hat{\epsilon}^{(f)}_{i,m}$ in the update formulas of (9)-(10) around the mean of the smoothing densities. In other words, $\hat{\epsilon}^{(f)}_{i,m} \approx \hat{\epsilon}^{(f)}_{i,m}$ where $\hat{\epsilon}^{(f)}_{i,m}$ is computed similarly to (8) with $\hat{\pi}^{(f)}_{(i,j),m}$'s evaluated at $X_1^N = \bar{x}_{1:N} := E_x\{x_{1:N}\}$ and $\Theta = \hat{\Theta}^{(f)}$ in Eq. (6).

**Remark 2.** Note that this approximation is valid when the GM smoothing densities (over which the expectations are computed) do not exhibit multimodal behavior with mixture components far from each other. Otherwise, the 0th order approximation must be carried out at the mean of each mixture component separately (rather than at the mean of the smoothing density). Under high enough observation signal-to-noise ratio...
(SNR), the GM smoothing densities are expected to mainly consist of mixture components with similar means, so the resulting density exhibits a unimodal morphology concentrated on the ML estimate of the states. Therefore, approximation of $\hat{\epsilon}_{i,m}(\ell)$ by its value at the mean of the smoothing density would not introduce significant error at high SNRs. It is worth noting that higher order approximations to $\hat{\epsilon}_{i,m}(\ell)$ can be considered at the cost of more computational cost, which would also result in higher moments of GM smoothing densities appearing in the M-step update equations. As we will demonstrate in the Results section, the 0th order approximation suffices for our applications.

It is known that for a linear SSM with Gaussian mixture noise, the filtering and smoothing densities also take Gaussian mixture forms [5,9]. Let $P \left( X_{n-1} \mid Y_1^N, \tilde{\Theta}(\ell) \right) = \sum_{\gamma=1}^{\Gamma_S} \rho_n^{(s,\gamma)} N \left( x_n; \mu_n^{(s,\gamma)}, \Sigma_n^{(s,\gamma)} \right)$ be the one-step joint smoothing density at time $n$, where the superscript $s$ identifies smoothing parameters, and $\Gamma_S$ is the number of mixture components forming the smoothing density. Taking $\bar{\epsilon}_{i,m}(\ell)$ out of the expectations, the M-step update equations become:

$$\hat{p}_m^{(\ell+1)} = \frac{1}{K} \sum_{i=1}^{K} \epsilon_i^{(\ell)}_{i,m}, \quad \hat{\mu}_m^{(\ell+1)} = \frac{1}{W} \sum_{i=1}^{K} \sum_{j=1}^{W} A_{(i,j)} \sum_{\gamma=1}^{\Gamma_S} \rho_n^{(s,\gamma)} \left( \Sigma_n^{(s,\gamma)} + \mu_n^{(s,\gamma)} \left( \mu_n^{(s,\gamma)} \right)^\top \right) \tilde{A}_{(i,j)} - \hat{p}_m^{(\ell+1)} \hat{\mu}_m^{(\ell+1)^\top}, \quad \hat{S}_m^{(\ell+1)} = \frac{1}{W} \sum_{i=1}^{K} \epsilon_i^{(\ell)}_{i,m} \tilde{A}_{(i,j)} \sum_{\gamma=1}^{\Gamma_S} \rho_n^{(s,\gamma)} \left( \Sigma_n^{(s,\gamma)} + \mu_n^{(s,\gamma)} \left( \mu_n^{(s,\gamma)} \right)^\top \right) \tilde{A}_{(i,j)} - \hat{p}_m^{(\ell+1)} \hat{\mu}_m^{(\ell+1)^\top},$$

where $\tilde{A}_{(i,j)} = [-A_{(i,j)}, I_{d_x}]$ with $I_{d_x}$ denoting the identity matrix of dimension $d_x$. Note that the update rule for $R$ in Eq. (11) takes the form:

$$\hat{R}^{(\ell+1)} = \frac{1}{N} \sum_{n=1}^{N} y_n y_n^\top - \frac{K}{M} \sum_{m=1}^{M} \sum_{i,m}^{K} \epsilon_i^{(\ell)} \sum_{j=1}^{W} \sum_{\gamma=1}^{\Gamma_S} y_{(i,j)} \left( \mu_{(i,j)}^{(s,\gamma)} \right)^\top C_{\alpha}^\top + C_{\alpha} \left( \mu_{(i,j)}^{(s,\gamma)} \right) y_{(i,j)}^\top - C_{\alpha} \left( \Sigma_{(i,j)}^{(s,\gamma)} \right) C_{\alpha}^\top.$$

Similarly, the coordinate descent update rule for $\alpha$ in Eq. (12) takes the form:
\[ \alpha^{(\ell+1)} = \frac{\sum_{i=1}^{K} \sum_{m=1}^{M} \sum_{j=1}^{W} \sum_{\gamma=1}^{\Gamma_5} \text{Tr} \left\{ \sum_{m}^{(\ell+1)-1} \left( \sum^{(s,\gamma)}_{(i,j)} \right)_{1,1} \right\}}{\sum_{i=1}^{K} \sum_{m=1}^{M} \sum_{j=1}^{W} \sum_{\gamma=1}^{\Gamma_5} \text{Tr} \left\{ \sum_{m}^{(\ell+1)-1} \left( \sum^{(s,\gamma)}_{(i,j)} \right)_{2,1} - \left( \mu^{(s,\gamma)}_{(i,j)} \right)^{\top} \mu^{(\ell+1)}_{m} \right\}}. \]  

(S7)

where \((Z)_{a,b}\) and \((z)_{a}\) denote the \((a,b)\text{th}\) block and \(a\text{th}\) block of the matrix \(Z\) and vector \(z\), respectively. In the TRF model of Eq. 17, we have \(C_n = S_n^T \tilde{G}\).

Another approach to approximately compute the expectations in the update equations (9)-(10) is to use the Laplace approximation [10]. This approach, however, requires the computation of the GM joint smoothing density of \(X^{(i,W)}_{(i,0)} \mid Y_{1}^{N}, \hat{\Theta}^{(\ell)}\) and is more computationally intensive than the current approximation, which only requires the one-step smoothing covariances regardless of the choice of \(W\).

The smoothing density parameters in Eq. (S4), i.e., \(\{\rho^{(s,\gamma)}_{n, \mu^{(s,\gamma)}_{n}}, \Sigma^{(s,\gamma)}_{n}\}\), have to be estimated for \(n = 1, \ldots, N\) in the E-step. In Section II.D of [5], a forward-backward recursion is used to obtain closed-form solutions for smoothing densities under a linear SSM with GM noise components. The dimension of the underlying matrices and matrix inversion costs, however, grows with \(n\) as the recursions proceed, which limits the utility of the algorithm for practical applications even with moderate observation duration.

In [9], the two-filter formula is adopted to compute the GM smoothing densities by transforming the smoothing problem to a filtering one. An underlying assumption in [9] is that either \(C_n\) is invertible or consecutive observations can be concatenated such that the effective measurement matrix is invertible. As this assumption does not hold in general, we instead develop a recursive algorithm based on the two-filter formula in our setting to compute the smoothing parameters in (S4) in closed-form. Since all of the following densities are conditioned on \(\hat{\Theta}^{(\ell)}\) similar to Eq. (S4), we hereafter drop the conditioning in our notation for convenience.

Let the filtering density at the end of the \((i-1)\text{st}\) window be

\[ P \left( x_{(i,0)} \mid Y_{1}^{(i,0)} \right) = \sum_{\gamma=1}^{\Gamma_f} \rho^{(f,\gamma)}_{(i,0)} \mathcal{N} \left( x_{(i,0)} ; \mu^{(f,\gamma)}_{(i,0)}, \Sigma^{(f,\gamma)}_{(i,0)} \right), \]  

(S8)

where superscript \(f\) identifies forward filtering parameters and \(\Gamma_f\) is the number of mixtures forming the filtering density at the end of each window. Also, let the unnormalized backward information filter [9] at the end of the \(i\text{th}\) window be
filtering density
backward filter
joint smoothing density

Fig S1. Schematic depiction of forward/backward updates for Eq. (S10).

\[
\text{P}(\mathbf{x}_{1W}^N | \mathbf{x}_{iW}) \propto \sum_{\gamma=1}^{\Gamma_B} \rho_{\gamma W}^{(iW)} \exp \left\{ -\frac{1}{2} \mathbf{x}_{iW}^T \mathbf{B}_{iW}^{(\gamma)} \mathbf{x}_{iW} + \mathbf{x}_{iW}^T \mathbf{b}_{iW}^{(\gamma)} \right\}, \quad (S9)
\]
where \( \Gamma_B \) is the number of exponential components forming the information filter at the end of each window. Note that Eq. (S9) is not a density in \( \mathbf{x} \). Considering the independence of \( z_i \) and \( \mathbf{y}_{1}^{(i,0)} \), the two-filter formula for window \( i \) in our switching GM process noise model can be written as

\[
\text{P} \left( \mathbf{x}_{(i,j)}^{(i,j-1)} | \mathbf{y}_{1}^{(i,0)} \right) = \frac{1}{\text{P} \left( \mathbf{y}_{1}^{(i,0)} \right)} \sum_{m=1}^{M} \rho_{m}^{(f)} \text{P} \left( \mathbf{x}_{(i,j-1)}, \mathbf{y}_{1}^{(i,j-1)} | \mathbf{y}_{1}^{(i,0)}, s_i^m \right) \times \text{P} \left( \mathbf{x}_{(i,j)} | \mathbf{x}_{(i,j-1)}, s_i^m \right) \text{P} \left( \mathbf{y}_{1}^{(i,j)} | \mathbf{x}_{(i,j)}, s_i^m \right), \quad (S10)
\]
where \( s_i^m \) stands for the event \( \{ z_i = m \} \). Our main objective here is to find a recursive update rule for the joint density of Eq. (S10). To this end, we derive recursive formulas for propagating the forward and backward filters, i.e., the components in the summand of Eq. (S10), within each window (See Fig. S1 for a schematic depiction of this procedure).

The leftmost term in the summand of Eq. (S10) is the forward filter and represents an unnormalized filtering density, which we express as:

\[
\text{P} \left( \mathbf{x}_{(i,j)}, \mathbf{y}_{1}^{(i,j)} | \mathbf{y}_{1}^{(i,0)}, s_i^m \right) = \sum_{\gamma=1}^{\Gamma_F} \rho_{(i,j),m}^{(f,\gamma)} \mathcal{N} \left( \mathbf{x}_{(i,j)}; \mu_{(i,j),m}^{(f,\gamma)}, \Sigma_{(i,j),m}^{(f,\gamma)} \right) \quad (S11)
\]
for \( j = 1, \ldots, W \) and compute it through the following unnormalized forward recursion in \( j \):
\[
P(x_{i,j}, \mathbf{y}^{(i,j)}_{(i,1)} \mid \mathbf{y}_{1}^{(i,0)}, \mathbf{y}_{1}^{(i,1)}, \mathbf{y}_{1}^{(i,2)}) = \int P(y_{i,j} \mid x_{i,j}) P(x_{i,j} \mid x_{i,j-1}, y_{i,j-1}^{(i,1)}) \times \notag\]
\[
P(x_{i,j-1}, \mathbf{y}^{(i,j-1)}_{(i,1)} \mid \mathbf{y}_{1}^{(i,0)}, \mathbf{y}_{1}^{(i,1)}, \mathbf{y}_{1}^{(i,2)}) \, dx_{i,j-1}.
\]

The recursion is initialized by the filtering density in Eq. (S8) at window \(i\). This results in the following forward filter parameter updates:

\[
\begin{align*}
\bar{\mu} &= A_{(i,j)} \mu_{(i,j-1),m} + \tilde{\mu}^{(f)} \\
\bar{\Sigma} &= A_{(i,j)} \Sigma^{(f)}_{(i,j-1),m} A_{(i,j)}^{T} + \tilde{\Sigma}_{m} \\
H &= \bar{\Sigma} C_{(i,j)} (C_{(i,j)} \bar{\Sigma} C_{(i,j)}^{T} + R)^{-1} \\
\mu^{(f)}_{(i,j),m} &= \bar{\mu} + H (y_{(i,j)} - C_{(i,j)} \bar{\mu}) \\
\Sigma^{(f)}_{(i,j),m} &= (I - HC_{(i,j)}) \tilde{\Sigma} \\
\rho^{(f)}_{(i,j),m} &= \rho^{(f)}_{(i,j-1),m} N \left( y_{(i,j)}; C_{(i,j)} \mu_{(i,j),m} + C_{(i,j)} \Sigma_{C_{(i,j)}} + R \right)
\end{align*}
\]

and the filtering density at time \((i,j)\) is computed as

\[
P(x_{i,j} \mid \mathbf{y}_{1}^{(i,j)}) \propto \sum_{m=1}^{M} \tilde{\rho}_{m} P(x_{i,j}, \mathbf{y}_{1}^{(i,j)} \mid \mathbf{y}_{1}^{(i,0)}, \mathbf{y}_{1}^{(i,1)}).
\]

Next, we represent the unnormalized backward information filter \(P(\mathbf{y}^{N}_{(i,j)} \mid x_{(i,j)}, \mathbf{y}_{m}^{(i)})\) in Eq. (S10), as

\[
\sum_{\gamma=1}^{\Gamma_{B}} \beta_{(i,j),m}^{(\gamma)} \exp \left\{-\frac{1}{2} x^{T}_{(i,j)} B_{(i,j),m} x_{(i,j)} + x^{T}_{(i,j)} b_{(i,j),m}^{(\gamma)} \right\},
\]

where we enforce the normalization \(\sum_{\gamma=1}^{\Gamma_{B}} \sum_{m=1}^{M} \beta_{(i,j),m}^{(\gamma)} = 1\). Note that this normalization is applied to avoid numerical instabilities while performing the recursions and does not change the final smoothing density of Eq. (S10), which has to be eventually normalized. The backward filter in Eq. (S14) can be computed through the following recursion \[9\]:

\[
P(\mathbf{y}^{N}_{(i,j)} \mid x_{(i,j)}, \mathbf{y}_{1}^{(i)}) = \int P(y_{i,j} \mid x_{i,j}) P(x_{i,j+1} \mid x_{i,j}, \mathbf{y}_{1}^{(i)}) \times \notag\]
\[
P(\mathbf{y}_{1}^{N} \mid x_{(i,j+1)}, \mathbf{y}_{1}^{(i)}) \, dx_{i,j+1}.
\]
and is initialized by Eq. [S9] at the end of window $i$. This results in the following backward filter parameter updates:

$$
\begin{align*}
\hat{\Sigma} &= \hat{\Sigma}^{(f)}(t) \left( I + B^{(\gamma)}(t, i, j+1, m) \hat{\Sigma}^{(f)}(t) \right) \\
\hat{\mu} &= \hat{\Sigma}^{(f)}(t) B^{(\gamma)}(t, i, j+1, m) + \hat{\mu}^{(f)}(t) \\
B^{(\gamma)}(t, i, j+1, m) &= C^{(t)}(i, j) R^{-1} C^{(t)}(i, j+1) + A^{(t)}(i, j+1) \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} - \hat{\Sigma}^{(f)}(t) \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} A^{(t)}(i, j+1) \\
b^{(\gamma)}(t, i, j+1, m) &= C^{(t)}(i, j) R^{-1} y^{(t)} - A^{(t)}(i, j+1) \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} \hat{\mu}^{(f)}(t) + A^{(t)}(i, j+1) \hat{\Sigma}^{(f)}(t) \hat{\mu}^{(f)}(t) \\
\beta^{(\gamma)}(t, i, j+1, m) &= \exp \left\{ -\frac{1}{2} \left( \hat{\mu}^{(f)}(t) \right)^{T} \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} \hat{\mu}^{(f)}(t) + \frac{1}{2} \hat{\mu}^{(f)}(t)^{T} \hat{\Sigma}^{(f)}(t) \hat{\mu}^{(f)}(t) \right\}
\end{align*}
$$

(S16)

and the overall backward filter in the beginning of window $i$ can be computed from Eq. [S14] as

$$
P \left( y^{(N)}(i, 0) \mid x^{(i, 0)} \right) = \sum_{m=1}^{M} \beta^{(f)}(t, i, m) P \left( y^{(N)}(i, 0) \mid x^{(i, 0)}, \beta^{(m)} \right) .
$$

(S17)

Using Eqs. [S11] and [S14], the parameters of the joint GM smoothing density in Eq. [S10] are computed as:

$$
\begin{align*}
\gamma'' &= (\gamma - 1) M \Gamma_{B} + (m - 1) \Gamma_{B} + \gamma' \\
S_{11} &= \left( \Sigma^{(f, m)}(i, j-1, \gamma) \right)^{-1} + A^{(t)}(i, j) \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} A^{(t)}(i, j) \\
S_{12} &= S_{21}^{T} = -A^{(t)}(i, j) \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} \\
S_{22} &= B^{(m)}(i, j, \gamma') + \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} \\
\Sigma^{(s)}(i, j, \gamma'') &= \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \\
\mu^{(s)}(i, j, \gamma'') &= \begin{bmatrix} \mu^{(s)}(i, j, \gamma') \end{bmatrix} \\
\rho^{(s)}(i, j, \gamma'') &= \mu^{(s)}(i, j, \gamma') \beta^{(m)}(i, j, \gamma') \sqrt{\frac{||\Sigma^{(s)}(i, j, \gamma'')||}{||\Sigma^{(f, m)}(i, j, \gamma')||}} \\
&\times \exp \left\{ -\frac{1}{2} \left( \mu^{(s)}(i, j-1, \gamma') \right)^{T} \left( \Sigma^{(f, m)}(i, j-1, \gamma) \right)^{-1} \mu^{(s)}(i, j-1, \gamma') \right\} \\
&\times \exp \left\{ -\frac{1}{2} \left( \mu^{(f)}(t, i, j+1, m) \right)^{T} \left( \hat{\Sigma}^{(f)}(t) \right)^{-1} \mu^{(f)}(t, i, j+1, m) + \frac{1}{2} \left( \mu^{(s)}(i, j, \gamma'') \right)^{T} \left( \Sigma^{(s)}(i, j, \gamma'') \right)^{-1} \mu^{(s)}(i, j, \gamma'') \right\}
\end{align*}
$$

(S18)
where we have $\gamma \in \{1, \ldots, \Gamma_F\}$, $m \in \{1, \ldots, M\}$, and $\gamma' \in \{1, \ldots, \Gamma_B\}$. This brings the total number of mixture components in the joint smoothing density of Eq. (S10) to $\Gamma_F \times M \times \Gamma_B$. As the number of mixture components grows exponentially in SSMs with GM noise components [11], limiting them is a crucial step for practical purposes. To this end, in forming the density of Eq. (S4), the number of mixture components obtained from Eq. (S10) is reduced to $\Gamma_S$ prior to updating the parameters in the M-step. In this work, we choose $\Gamma_S$ components from the density of Eq. (S10) with the largest mixture probabilities for simplicity. However, more accurate mixture reduction algorithms are available and developed in [11–13], but with additional computational costs. It is worth noting that calculations corresponding to the weights $\rho^{(i)}$’s in Eq. (S12), $\beta$’s in (S16), and $\rho^{(v)}$’s in (S18) should be performed in log-scale to avoid numerical errors in practice.

Algorithm S2. Two-Filter Gaussian Mixture Smoothing Alg.

**Input:** linear state-space model in (1), parameter estimate $\hat{\Theta}^{(i)}$, and component limits $\Gamma_F$, $\Gamma_B$, and $\Gamma_S$.

**Output:** smoothing density parameters $\rho^{(i)}_{n,\gamma}$, $\mu^{(i)}_{n,\gamma}$, $\Sigma^{(i)}_{n,\gamma}$ in (S4) for $n \in \{1, \ldots, N\}$ and $\gamma \in \{1, \ldots, \Gamma_S\}$.

1: Initialize the filtering density in (S8) at $n = 0$ as the prior on $x_0$.
2: for $i = 1 : K$ do
3: Run forward recursions of (S12) for $m = 1, \ldots, M$ in window $i$ starting from (S8) and store the parameters.
4: Compute the filtering density at $n = iW$ from (S13).
5: Out of $\Gamma_F \times M$ mixture components in the filtering density, keep $\Gamma_F$ with the largest probabilities as initialization for window $i + 1$.
6: end for
7: Initialize the backward filter as $P(y_N | x_N)$, i.e., $\beta_{N,1} = 1$, $B_{N,1} = C_N^T R^{-1} C_N$, and $b_{N,1} = C_N^T R^{-1} y_N$.
8: for $i = K : 1$ do
9: Run backward recursions of (S16) for $m = 1, \ldots, M$ in window $i$ starting from (S9).
10: Run smoothing algorithm of (S18) in window $i$ using backward filtering parameters and the stored forward filtering parameters.
11: Out of $\Gamma_F \times M \times \Gamma_B$ smoothing mixture components, store $\Gamma_B$ with the largest probabilities for smoothing densities of (S4) in window $i$.
12: Compute the backward filter at $n = (i - 1)W$ from (S17).
13: Out of $\Gamma_B \times M$ backward filtering components, keep $\Gamma_B$ corresponding to the most significant mixture components of $P(X^{(i,1)}_{(i,0)} | Y^N_1)$ as initialization for window $i - 1$.
14: end for
15: Output the computed smoothing parameters of (S4).
Model Order Selection

An important issue in applications of GMs for clustering is the choice of the number of mixtures \( M \). Various model selection criteria have been used in the literature of Gaussian mixtures, including the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Integrated Completed Likelihood (ICL) \([14-17]\), most of which require the computation of data log-likelihood. In Approach 1, the log-likelihood can be approximated using the unnormalized particle filtering weights \( \tilde{\omega}_i^{(u)} \)'s in Algorithm S1 as

\[
\log P (\mathcal{Y}_1^N) \approx \sum_{i=1}^{K} \log \left( \sum_{n=1}^{U} \tilde{\omega}_i^{(u)} \right). 
\] (S19)

In Approach 2, using the unnormalized filtering densities in closed-form approximation, the log-likelihood in our model can be computed based on \([18]\) as

\[
\log P (\mathcal{Y}_1^N) = \sum_{i=1}^{K} \log \left( \sum_{m=1}^{M} \hat{p}^{(\ell)}_m P \left( \mathcal{Y}_1^{(i,W)} \mid \mathcal{Y}_1^{(i,0)}, \gamma_i^m \right) \right) \\
\approx \sum_{i=1}^{K} \log \left( \sum_{m=1}^{M} \hat{p}^{(\ell)}_m \left( \sum_{\gamma=1}^{V_f} \rho^{(\ell,\gamma)}_i \right) \right) 
\] (S20)

where the last line is derived from integrating the unnormalized filtering density in Eq. (S11).

Details of the MSAR Estimation Procedure

As in the case of SSMs with GM process noise, we consider a Markov switching process in which the states are constant over consecutive windows of length \( W \) indexed by \( k = 1, 2, \ldots, K \), where \( K := N/W \). Recall that the parameters to be estimated are \( \mathcal{M} := \left\{ \{ \pi_i \}_{i=1}^{J}, \{ P_{ij} \}_{i,j=1}^{J,J}, \{ \alpha_j, \mu_j, Q_j \}_{j=1}^{J} \right\} \). For simplicity, we assume that \( Q_j \) is diagonal. For notational convenience, we denote the Gaussian density with mean \( \mu \) and covariance \( Q \) evaluated at \( x \) by \( \mathcal{N}(x; \mu, Q) \). Finally, we denote \( \hat{X}_{n,(k-1)W+l}^{(i)} \) by \( \hat{X}_{(i,j)}^{(k,l)} \).

The parameters can be estimated using an instance of the EM algorithm as follows:
The E-step

Let \( \mathcal{M}^{(\ell)} := \left\{ \{ \pi_i^{(\ell)} \}_{i=1}^J, \{ P_{ij}^{(\ell)} \}_{i,j=1}^J, \{ a_j^{(\ell)}, \mu_j^{(\ell)}, Q_j^{(\ell)} \}_{j=1}^J \right\} \) be the parameter estimates at iteration \( \ell \). We define:

\[
\varepsilon_{j,k}^{(\ell)} := P \left[ s_k = j \middle| \tilde{X}_{(1,1)}^{(K,W)}, \mathcal{M}^{(\ell)} \right], \quad \xi_{i,j,k}^{(\ell)} := P \left[ s_k = j, s_{k-1} = i \middle| \tilde{X}_{(1,1)}^{(K,W)}, \mathcal{M}^{(\ell)} \right].
\] (S21)

It can be shown that:

\[
\varepsilon_{j,k}^{(\ell)} = \frac{a_{j,k}^{(\ell)} b_{j,k}^{(\ell)}}{\sum_{j=1}^J a_{j,k}^{(\ell)} b_{j,k}^{(\ell)}}, \quad \xi_{i,j,k}^{(\ell)} = \frac{a_{i,k-1}^{(\ell)} P_{ij}^{(\ell)} b_{j,k}^{(\ell)} \prod_{w=1}^W \mathcal{N}(\hat{x}_{(k,w)}^{(\ell)} - \alpha_j^{(\ell)} \hat{x}_{(k,w-1)}^{(\ell)}; \mu_j^{(\ell)}, Q_j^{(\ell)})}{\sum_{i=1}^J \sum_{m=1}^J a_{i,k-1}^{(\ell)} P_{im}^{(\ell)} b_{m,k}^{(\ell)} \prod_{w=1}^W \mathcal{N}(\hat{x}_{(k,w)}^{(\ell)} - \alpha_m^{(\ell)} \hat{x}_{(k,w-1)}^{(\ell)}; \mu_m^{(\ell)}, Q_m^{(\ell)})},
\] (S22)

where \( a_{j,k}^{(\ell)} \) and \( b_{j,k}^{(\ell)} \) are the forward and backward filters, respectively, defined as:

\[
a_{j,k}^{(\ell)} := P \left[ s_k = j, \tilde{X}_{(1,1)}^{(K,W)} \middle| \mathcal{M}^{(\ell)} \right], \quad b_{j,k}^{(\ell)} := P \left[ \tilde{X}_{(1,1)}^{(K,W)} \middle| s_k = j, \tilde{X}_{(k+1,1)}^{(K,W)} \middle| \mathcal{M}^{(\ell)} \right].
\] (S23)

The forward and backward filters can be recursively computed via the following recursions:

\[
a_{j,k}^{(\ell)} = \sum_{i=1}^J a_{i,k-1}^{(\ell)} P_{ij}^{(\ell)} \prod_{w=1}^W \mathcal{N}(\hat{x}_{(k,w)}^{(\ell)} - \alpha_j^{(\ell)} \hat{x}_{(k,w-1)}^{(\ell)}; \mu_j^{(\ell)}, Q_j^{(\ell)}),
\] (S24)

with initialization \( a_{j,1}^{(\ell)} = \pi_j^{(\ell)} \prod_{w=1}^W \mathcal{N}(\hat{x}_{(1,w)}^{(\ell)} - \alpha_j^{(\ell)} \hat{x}_{(1,w-1)}^{(\ell)}; \mu_j^{(\ell)}, Q_j^{(\ell)}) \), \( j = 1, 2, \ldots, J \).

And,

\[
b_{j,k}^{(\ell)} = \sum_{i=1}^J b_{i,k+1}^{(\ell)} P_{ji}^{(\ell)} \prod_{w=1}^W \mathcal{N}(\hat{x}_{(k+1,w)}^{(\ell)} - \alpha_i^{(\ell)} \hat{x}_{(k+1,w-1)}^{(\ell)}; \mu_i^{(\ell)}, Q_i^{(\ell)}),
\] (S25)

with initialization \( b_{j,K}^{(\ell)} = 1 \), \( j = 1, 2, \ldots, J \).

**Remark 3.** Note that a direct implementation of the recursions in Eqs. (S24) and (S25) is likely to result in numerical instabilities, given that the Gaussian density product generates notably small values. Given that the expressions for \( \varepsilon_{j,k}^{(\ell)} \) and \( \xi_{i,j,k}^{(\ell)} \) are scale invariant with respect to \( a_{j,k}^{(\ell)}, a_{j,k-1}^{(\ell)} \) and \( b_{j,k}^{(\ell)} \), one can implement the recursions of Eqs. (S24) and (S25) in an *unnormalized* fashion by rescaling the forward and backward filters after each iteration as:

\[
a_{j,k}^{(\ell)} \leftarrow \frac{a_{j,k}^{(\ell)}}{\sum_{i=1}^J a_{i,k}^{(\ell)}}, \quad b_{j,k}^{(\ell)} \leftarrow \frac{b_{j,k}^{(\ell)}}{\sum_{i=1}^J b_{i,k}^{(\ell)}}.
\] (S26)
It is straightforward to verify that Eq. (S22) remains unchanged as a result of this rescaling procedure. The log-sum-exp approximation can also be used to further avoid numerical instabilities in computing the Gaussian density products [19].

The M-step

The Markov chain parameters can be updated as follows:

\[
\pi_j^{(\ell+1)} = \frac{\epsilon_j^{(\ell)}}{\sum_{i=1}^J \epsilon_i^{(\ell)}}, \quad P_{ij}^{(\ell+1)} = \frac{\sum_{k=2}^K \zeta_{i,j,k}^{(\ell)}}{\sum_{k=1}^K \sum_{l=1}^J \zeta_{i,l,k}^{(\ell)}}. \tag{S27}
\]

The state-space parameters can be updated via coordinate descent as follows:

\[
\begin{align*}
\left( \mu_j^{(\ell+1)} \right)_m & = \frac{\sum_{k=1}^K \epsilon_{j,k}^{(\ell)} \sum_{w=1}^W \left( (\hat{x}(k,w))_m - \alpha_j^{(\ell)}(\hat{x}(k,w-1))_m \right)}{W \sum_{k=1}^K \epsilon_{j,k}^{(\ell)}}, \tag{S28}

\left( Q_j^{(\ell+1)} \right)_{m,m} & = \frac{\sum_{k=1}^K \epsilon_{j,k}^{(\ell)} \sum_{w=1}^W \left( (\hat{x}(k,w))_m - \alpha_j^{(\ell)}(\hat{x}(k,w-1))_m - (\mu_j^{(\ell+1)})_m \right)^2}{W \sum_{k=1}^K \epsilon_{j,k}^{(\ell)}}, \tag{S29}
\end{align*}
\]

for \( m = 1, 2, \cdots, d_x, \ j = 1, 2, \cdots, J \), and

\[
\alpha_j^{(\ell+1)} = \frac{\sum_{k=1}^K \epsilon_{j,k}^{(\ell)} \sum_{m=1}^{d_x} \frac{1}{Q_j^{(\ell+1)}} \sum_{w=1}^W (\hat{x}(k,w-1))_m \left( (\hat{x}(k,w))_m - (\mu_j^{(\ell+1)})_m \right)}{\sum_{k=1}^K \epsilon_{j,k}^{(\ell)} \sum_{m=1}^{d_x} \frac{1}{Q_j^{(\ell+1)}} (\hat{x}(k,w-1))_m^2}. \tag{S30}
\]

The EM algorithm proceeds until a convergence criterion is met at iteration \( L_0 \), upon which the MSAR estimates can be computed as:

\[
\hat{x}_{(k,w)}^{(\text{MSAR})} := \sum_{j=1}^J \omega_{j,k}^{(L_0)} \left( \mu_j^{(L_0)} + \alpha_j^{(L_0)} \hat{x}(k,w-1) \right), \quad n = 1, 2, \cdots, N, \tag{S31}
\]

where

\[
\omega_{j,k}^{(L_0)} := P \left[ s_k = j \bigg| \hat{x}_{(1,1)}^{(1)}, \mathcal{M}^{(L_0)} \right] = \frac{a_{j,k}^{(L_0)}}{\sum_{i=1}^J a_{i,k}^{(L_0)}}. \tag{S32}
\]
References

1. Cappé O, Godsill SJ, Moulines E. An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE. 2007;95(5):899–924.

2. Olsson J, Cappé O, Douc R, Moulines E, et al. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. Bernoulli. 2008;14(1):155–179.

3. Klaas M, Briers M, De Freitas N, Doucet A, Maskell S, Lang D. Fast particle smoothing: If I had a million particles. In: Proceedings of the 23rd International Conference on Machine Learning. ACM; 2006. p. 481–488.

4. Fearnhead P, Wynoll D, Tawn J. A sequential smoothing algorithm with linear computational cost. Biometrika. 2010;97(2):447–464.

5. Vo BN, Vo BT, Mahler RP. Closed-form solutions to forward–backward smoothing. IEEE Trans on Sig Proc. 2012;60(1):2–17.

6. Anderson BD, Moore JB. Optimal filtering. Courier Corporation; 2012.

7. Wan EA, Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium. IEEE; 2000. p. 153–158.

8. Gibson S, Ninness B. Robust maximum-likelihood estimation of multivariable dynamic systems. Automatica. 2005;41(10):1667–1682.

9. Kitagawa G. The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. Annals of the Institute of Statistical Mathematics. 1994;46(4):605–623.

10. Wong R. Asymptotic approximations of integrals. SIAM; 2001.

11. Sorenson HW, Alspach DL. Recursive Bayesian estimation using Gaussian sums. Automatica. 1971;7(4):465–479.

12. Runnalls AR. Kullback-Leibler approach to Gaussian mixture reduction. IEEE Trans on Aero and Elec Sys. 2007;43(3):989–999.
13. Crouse DF, Willett P, Pattipati K, Svensson L. A look at Gaussian mixture reduction algorithms. In: 14th International Conference on Information Fusion. IEEE; 2011. p. 1–8.

14. Ding J, Tarokh V, Yang Y. Model selection techniques: An overview. IEEE Sig Proc Mag. 2018;35(6):16–34.

15. Figueiredo MAT, Jain AK. Unsupervised learning of finite mixture models. IEEE Trans on Patt An & Mach Int. 2002;(3):381–396.

16. Biernacki C, Celeux G, Govaert G. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans on Patt An & Mach Int. 2000;22(7):719–725.

17. Mehrjou A, Hosseini R, Araabi BN. Improved Bayesian information criterion for mixture model selection. Pattern Recognition Letters. 2016;69:22–27.

18. De Jong P. The likelihood for a state space model. Biometrika. 1988;75(1):165–169.

19. Boyd S, Vandenberghe L. Convex optimization. Cambridge University Press; 2004.