ABSTRACT

Title of dissertation: Efficient Solutions to
High-Dimensional and Nonlinear
Neural Inverse Problems

Sayyed Sina Miran
Doctor of Philosophy, 2019

Dissertation directed by: Professor Behtash Babadi
Department of Electrical and Computer Engineering

Development of various data acquisition techniques has enabled researchers to study the brain as a complex system and gain insight into the high-level functions performed by different regions of the brain. These data are typically high-dimensional as they pertain to hundreds of sensors and span hours of recording. In many experiments involving sensory or cognitive tasks, the underlying cortical activity admits sparse and structured representations in the temporal, spatial, or spectral domains, or combinations thereof. However, current neural data analysis approaches do not take account of sparsity in order to harness the high-dimensionality. Also, many existing approaches suffer from high bias due to the heavy usage of linear models and estimation techniques, given that cortical activity is known to exhibit various degrees of non-linearity. Finally, the majority of current methods in computational neuroscience are tailored for static estimation in batch-mode and offline settings, and with the advancement of brain-computer interface technologies, these methods need to be extended to capture neural dynamics in a real-time fashion. The

objective of this dissertation is to devise novel algorithms for real-time estimation settings and to incorporate the sparsity and non-linear properties of brain activity for providing efficient solutions to neural inverse problems involving high-dimensional data. Along the same line, our goal is to provide efficient representations of these high-dimensional data that are easy to interpret and assess statistically.

First, we consider the problem of spectral estimation from binary neuronal spiking data. Due to the non-linearities involved in spiking dynamics, classical spectral representation methods fail to capture the spectral properties of these data. To address this challenge, we integrate point process theory, sparse estimation, and non-linear signal processing methods to propose a spectral representation modeling and estimation framework for spiking data. Our model takes into account the sparse spectral structure of spiking data, which is crucial in the analysis of electrophysiology data in conditions such as sleep and anesthesia. We validate the performance of our spectral estimation framework using simulated spiking data as well as multi-unit spike recordings from human subjects under general anesthesia.

Next, we tackle the problem of real-time auditory attention decoding from electroencephalography (EEG) or magnetoencephalography (MEG) data in a competing-speaker environment. Most existing algorithms for this purpose operate offline and require access to multiple trials for a reliable performance; hence, they are not suitable for real-time applications. To address these shortcomings, we integrate techniques from state-space modeling, Bayesian filtering, and sparse estimation to propose a real-time algorithm for attention decoding that provides robust, statistically interpretable, and dynamic measures of the attentional state of the lis-

tener. We validate the performance of our proposed algorithm using simulated and experimentally-recorded M/EEG data. Our analysis reveals that our algorithms perform comparable to the state-of-the-art offline attention decoding techniques, while providing significant computational savings.

Finally, we study the problem of dynamic estimation of Temporal Response Functions (TRFs) for analyzing neural response to auditory stimuli. A TRF can be viewed as the impulse response of the brain in a linear stimulus-response model. Over the past few years, TRF analysis has provided researchers with great insight into auditory processing, specially under competing speaker environments. However, most existing results correspond to static TRF estimates and do not examine TRF dynamics, especially in multi-speaker environments with attentional modulation. Using state-space models, we provide a framework for a robust and comprehensive dynamic analysis of TRFs using single trial data. TRF components at specific lags may exhibit peaks which arise, persist, and disappear over time according to the attentional state of the listener. To account for this specific behavior in our model, we consider a state-space model with a Gaussian mixture process noise, and devise an algorithm to efficiently estimate the process noise parameters from the recorded M/EEG data. Application to simulated and recorded MEG data shows that the proposed state-space modeling and inference framework can reliably capture the dynamic changes in the TRF, which can in turn improve our access to the attentional state in competing-speaker environments.

Efficient Solutions to
High-Dimensional and Nonlinear Neural Inverse Problems

by

Sayyed Sina Miran

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Behtash Babadi, Chair/Advisor
Professor Jonathan Z. Simon
Professor Shihab Shamma
Professor Steven I. Marcus
Professor Michael C. Fu, Dean's Representative

Dedication

To *maman* and *baba* for their unconditional love and support

Acknowledgments

I would like to thank my advisor, Prof. Behtash Babadi, for his guidance and support in the course of my PhD, and to acknowledge his technical contribution to this thesis. His enthusiasm for research, bright ideas, accessibility for discussions, and in-depth knowledge of signal processing algorithms were crucial in the completion of this dissertation. I am also grateful to my committee members Prof. Jonathan Simon, Prof. Shihab Shamma, Prof. Steve Marcus, and Prof. Michael Fu for their constructive feedback and valuable suggestions on this thesis and, more importantly, during our collaborations. I am specially thankful to Prof. Simon for his inputs to Chapters 3 and 4 of this thesis. His careful paper revisions and vast knowledge of auditory neuroscience have significantly improved the quality of our papers together. Chapter 3 of this thesis is the result of a collaboration with Dr. Tao Zhang's group at Starkey Hearing Technologies. Tao has been a very caring and supportive supervisor, and I would like to thank him for this invaluable opportunity.

I was fortunate to have excellent group members throughout my PhD. I am specially grateful to Dr. Sahar Akram, Dr. Alireza Sheikhattar, and Dr. Abbas Kazemipour, with whom I have had fruitful collaborations and from whom I have learned a lot. I also want to express my gratitude to Dr. Alessandro Presacco for providing me with his hard-earned and unique datasets to work on. During my PhD coursework, I had the pleasure of learning various subjects from the knowledgeable professors at UMD. Among them, special appreciations go to Prof. Tom Goldstein for numerical optimization, Prof. Prakash Narayan for estimation theory, Prof.

Michael Rotkowitz for convex optimization, and Prof. Piya Pal for sparse statistical signal processing. I also would like to thank Melanie Prange, Vivian Lu, Heather Stewart, and Bill Churma as the staff of the ECE department at UMD who have been extremely friendly and helpful to me over the past five years.

I was blessed to have wonderful friends in the DC area throughout my time at UMD, and I am grateful to them for making my graduate studies such an enjoyable experience. A very special thank you goes to my high school and college friends, who are among my dearest and oldest friends. Although, at times, life has taken us thousands of miles away from each other, our friendship has remained as strong as ever. I am deeply thankful to the newly minted Dr. Anahita Abazari for her love and support in the past two years. I could not have asked for a more caring and understanding partner during the stressful parts of my PhD.

Finally and most importantly, I owe my parents, Manijeh and Mahmoud, a great debt of gratitude for all their selfless love and support over the years. You have encouraged me to shoot for the stars and have always provided me with the means to do so. This thesis is dedicated to you as a small thank you for all your sacrifices. Although I have always felt your presence and love close to my heart, the toughest part of my PhD has been not seeing you and spending time with you in the past few years because of an inhumane travel ban. I love you, and I miss you!

Table of Contents

List of Figures

List of Abbreviations

AR	Autoregressive
BCI	Brain-Computer Interface
CDF	Cumulative Distribution Function
CIF	Conditional Intensity Function
CS	Compressed Sensing
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTFT	Discrete-Time Fourier Transform
ECoG	Electrocorticography
EEG	Electroencephalography
EM	Expectation Maximization
FASTA	Fast Adaptive Shrinkage/Thresholding Algorithm
FBS	Forward-Backward Splitting
GLM	Generalized Linear Models
GM	Gaussian Mixture
LFP	Local Field Potential
MAP	Maximum A Posteriori
MEG	Magnetoencephalography
ML	Maximum Likelihood
MRAS	Model Reference Adaptive Search
MSE	Mean Squared Error
PDF	Probability Density Function
PSD	Power Spectral Density
PSTH	Peristimulus Time Histogram
RLS	Recursive Least Squares
SPARLS	Sparse RLS
SSM	State-Space Model
TRF	Temporal Response Function

Chapter 1: Introduction

Recent technological developments in neural data acquisition have provided researchers with abundant neural data sets with various spatiotemporal resolutions and recording modalities to study the brain. Neuroimaging techniques can be classified into invasive technologies, such as electrocorticography (ECoG), and noninvasive technologies, such as M/EEG and functional magnetic resonance imaging (fMRI). The high temporal resolutions of M/EEG ($\sim 1\,\mathrm{ms}$), convenient recording procedure, and widespread applications in Brain-Computer Interface (BCI) systems and neural prosthetics have increased the popularity of M/EEG in studying the human brain.

In order to relate the recorded data from the brain to the underlying functions, one needs to solve the so-called ”neural inverse problem”. The general goal in a neural inverse problem is to make inference on the activity of neuronal populations inside the brain or to decode the performed cognitive or sensory task, having observed the neuroimaging data. Existing algorithms for such problems, however, face the following key challenges in deciphering the underlying dynamic behavior of the brain.

First, in many experiments involving sensory or cognitive tasks, the underlying cortical domains responsible for information processing are relatively focal, structured, and sparse [1]. This feature has been observed in various domains such as the spectral, temporal, or spatial representations of neural data. On the other hand, neuroimaging data usually pertain to hundreds of sensors and can span hours of recording. Hence, neural data are relatively *high dimensional* with respect to the number of involved regions of the brain in typical experiments. For instance, fMRI data stream is approximately 1 million voxels per second, where each voxel corresponds to a cube of $1\,\mathrm{mm} \times 1\,\mathrm{mm} \times 1\,\mathrm{mm}$ in the brain [2]. An important challenge in computational neuroscience is, therefore, how to exploit the aforementioned sparse and structured representations in order to harness the high-dimensionality of neural data.

Second, it is known that there are various nonlinearities involved in cortical processing. In order to have easily interpretable results and to reduce the complexity of the models as well as their inference procedure, many existing approaches adopt linear models in processing neural data. However, this heavy usage of linear models and estimation techniques can induce large biases in the solutions themselves or the computed measures and features from them. Therefore, it is important to devise nonlinear models and estimation frameworks for neural inverse problems, which is motivated by the brain function itself.

Finally, most existing approaches for neural inverse problems involve offline processing and operate in batch-mode. In other words, they require either the whole duration of an experiment or multiple training trials to be available prior to processing. This type of processing and training data may not be possible or available in the emerging applications of neuroimaging such as BCI systems, neural prosthetics, and smart hearing aid devices, which require real-time processing of neural data with minimal amount of training data. Moreover, even in the batch-mode, most existing methods in computational neuroscience result in static estimates and do not provide a comprehensive dynamic picture of the brain function. Thus, the development of real-time and low-complexity algorithms for dynamic analysis of neuroimaging data is an important challenge to be addressed.

In this thesis, our goal is to design efficient algorithms for specific examples of the discussed high-dimensional and nonlinear neural inverse problems. At the same time, we consider efficient representations for the neural data, which are easily interpretable for diagnosis or soft-decision making purposes and effectively summarize the neuroimaging data for the task at hand. In Chapter 2, we consider the problem of devising a sparse spectral representation for binary neuronal spiking data, which is recorded invasively in the form of single- or multi-unit recordings. Neuronal spiking data exhibit sparse oscillatory components in their spectrum under conditions such as sleep or anesthesia [3,4]. Using the point process theory [5] and nonlinear signal processing techniques, we propose a new model for spectral representation of binary spiking data, which accounts for sparsity, and develop a fast procedure for its estimation. Chapter 3 considers the problem of real-time auditory attention decoding from M/EEG recordings in competing-speaker environments. Adopting techniques from Bayesian filtering, state-space modeling, and nonlinear signal processing, we develop a real-time algorithm for attention decoding in a dual-speaker setting which uses a minimal amount of training data for parameter tuning. The algorithm outputs a robust, statistically interpretable, and dynamic measure of the attentional state which can be used for soft-decision making in emerging applications such as smart hearing aid devices. In Chapter 4, we study the problem of dynamic TRF estimation, specially in competing-speaker environments, using state-space models. TRF components at specific lags may exhibit heterogeneous dynamics during the course of an experiment. These dynamics could be governed by the attentional state in a cocktail party setting. For instance, these components can include peaks that arise, persist, and disappear over time. To account for such dynamic behavior in our model, we consider a state-space model with Gaussian mixture process noise, where each mixture component captures a specific dynamic pattern. Then, we develop an algorithm to estimate the parameters of the Gaussian mixture process noise from single trial data. The Gaussian mixture process noise allows for reliable estimation of rapid changes in the TRF, which can enhance the utility of TRFs in attention decoding applications. Finally, we close this dissertation in Chapter 5 by concluding remarks regarding our contributions, discussing potential limitations, and outlining future directions of research.

Chapter 2: Robust Estimation of Sparse Narrowband Spectra from Binary Neuronal Spiking Data

Spectral analysis of time-series recorded from the brain, such as electroencephalography (EEG), has long been used for monitoring and characterizing brain activity in both clinical and research settings. Presence of specific oscillations in the EEG has been identified as the neural correlate of a variety of cognitive functions. Examples include the occipital alpha rhythms [6] and the somatomotor mu-rhythms [7]. Benefiting from the well-developed theory of spectral analysis of time-series, the spectral EEG signal processing techniques have been proven successful for diagnosis purposes such as the identification of epilepsy seizures [8, 9] and sleep disorders [10, 11].

Analysis of data from noninvasive recordings is limited by the low spatial resolution of the measurement mechanism, as the sensors record the integrated electrical activity of a large population of neurons in the brain. With the development of invasive recording procedures, acquisition of Local Field Potential (LFP) and single- and multi-unit recordings have also been made possible [12, 13]. The LFP captures the electrical activity of a more localized population of neurons compared to EEG, and single- and multi-unit recordings capture the neural activity at the neuronal level.

Although EEG signal processing techniques can be readily applied to LFP recordings, analysis of spike recordings has set forth various signal processing challenges due to their binary nature [14].

In recent years, the theory of point processes has been successfully employed to model and analyze binary spiking data [5, 15, 16]. These models provide a mathematically principled framework to relate the observed neuronal responses to the underlying covariates such as the sensory stimuli. In most of these applications, the point processes are used to model the neuronal responses in the time domain by enforcing temporal smoothness. The few exceptions which aim at calculating a frequency domain representation of the spiking data often proceed by computing an estimate of the spiking rate (as a continuous function) and then analyzing the spectral properties of the estimated rate. The spiking rate estimation techniques range from simple smoothing of the spiking histogram [17–19] to more sophisticated models which use generalized linear Gaussian state-space models to estimate the conditional intensity function (CIF) of the point process using Kalman filtering and smoothing techniques [20, 21]. The objective of these techniques is to provide a smoothed estimate of the spiking rate as a surrogate function whose power spectral density (PSD) is interpreted as the spectral representation of the spiking data. However, this interpretation has three immediate shortcomings. First, it is known that smoothing in the time domain results in blurring in the frequency domain [22], and hence these techniques are limited in terms of their spectral resolution. Second, spectral estimation requires estimating the second-order statistics of the underlying time-series, and even if the spiking rate is estimated accurately, the second-order statistics may not be. Third, these techniques are blind to the low-dimensional structure of neural data in conditions such as sleep [3], anesthesia [4], and epileptic seizures [23]. This low-dimensional structure is often manifested as sparsity in the spectral domain.

In this chapter, we address these shortcomings by casting the problem of spectral estimation from binary spiking data in the traditional discrete-parameter harmonic spectral estimation framework, where the objective is to estimate the second moments of a harmonic process driving the spiking activity. To this end, we model the spiking statistics of the underlying neurons by a conditional Bernoulli point process model, where the CIF is formed by mapping a stationary harmonic process through a logistic link function. Given the spiking data and considering sparsity-promoting priors, we compute the maximum *a posteriori* (MAP) estimate of the PSD of the harmonic process using the Expectation-Maximization (EM) algorithm. In addition, we construct confidence intervals for these estimates via sampling from the posterior distribution. Simulation studies concerning spiking data driven by sparse harmonic and autoregressive (AR) processes as well as application to real spiking data from anesthesia illustrate the superior performance of our proposed technique as compared to several existing techniques. Although our motivation stemmed from neuronal spiking data, it is worth noting that our modeling and estimation framework can be applied to any binary data modeled by point processes, such as the heart beat [24, 25], in order to extract a sparse spectral representation of the data.

The rest of this chapter is organized as follows: In Section 2.1, we introduce our model for the spiking activity of a population of neurons driven by a harmonic process. In Section 2.2, we derive the sparse MAP estimator of the PSD associated with the harmonic process. Section 2.2.3 discusses the construction of confidence intervals for the PSD estimate based on the Metropolis-Hastings sampling. Section 2.3 provides simulation results comparing our sparse PSD estimates with those obtained by existing methods for extracting the PSD of spiking data. Furthermore, we apply our estimator to real multi-unit recordings of spiking activity under general anesthesia. This is followed by our discussion and concluding remarks in Sections 2.4 and 2.5, respectively.

2.1 Preliminaries and Problem Formulation

Let $(0, T]$ be an observation interval during which the spiking activity of a neuron is recorded. For $t \in (0, T]$ let $N(t)$ be a point process representing the number of spikes in $(0, t]$ and H_t denote the spiking history in the interval $(0, t)$. We define the Conditional Intensity Function (CIF) of a point process $N(t)$ as [26]:

$$\lambda(t|H_t) := \lim_{\Delta \to 0} \frac{\mathrm{P}(N(t+\Delta) - N(t) = 1|H_t)}{\Delta} \tag{2.1}$$

In order to discretize the continuous-time point process, we consider bins of length Δ such that $T = K\Delta$, for some integer K. Assuming that Δ is small enough, the probability of having two or more spikes in an interval of Δ becomes negligible and the point process can be approximated in the k^{th} bin by a Bernoulli random variable n_k with success probability of $\lambda_k := \lambda\big(k\Delta|H_{k\Delta}\big)\Delta$, for $0 \leq k \leq K$. This assumption is biophysically plausible due to the absolute refractory period of neurons, and a choice of $\Delta \sim 1$ ms is typically sufficient to ensure that at most one spike occurs in any bin [5].

In general, oscillatory behavior of the neuronal spiking can be directly attributed to the oscillatory nature of the CIF. Our objective is to develop a method to estimate the PSD of the CIF from the observed binary spiking data. We consider an ensemble of L neurons driven by the same CIF, and denote the observed spike trains by $\{n_k^{(\ell)}\}_{\ell=1,k=1}^{L,K}$. The CIF, in turn, is modeled by a second-order stationary random process. We consider a simplified model where $\{x_k\}_{k=1}^{K}$ be a realization of the second-order stationary process with mean μ. In our model, we consider a logistic link for the CIF, such that $\lambda_k = \frac{1}{1+\exp(-x_k)}$. In summary, the model can be expressed as:

$$\begin{cases} \lambda_k = \dfrac{1}{1+e^{-x_k}}, \quad 1 \leq k \leq K \\ n_k^{(\ell)} \sim \text{Bernoulli}(\lambda_k), \quad 1 \leq k \leq K, 1 \leq \ell \leq L \end{cases} \tag{2.2}$$

and the objective is to estimate the PSD of x_k given the observations $\{n_k^{(\ell)}\}_{\ell=1,k=1}^{L,K}$.

In general, the PSD of second-order stationary processes can be characterized using the *Spectral Representation Theorem* [22]. This theorem implies that for the zero-mean, second-order stationary time series $x_k - \mu$ with spectral density function $S(\omega)$, there exists a continuous, orthogonal increment, and complex process $Z(\omega)$ such that

$$x_k - \mu = \int_{-\pi}^{\pi} e^{j\omega k} dZ(\omega) \tag{2.3}$$

where the integral is in the Riemann-Stieltjes sense and $E\{|dZ(\omega)|^2\} = S(\omega)d\omega$. The function $S(\omega)$ is referred to as the PSD. Several nonparametric estimation techniques, such as the Welch's method and multitaper estimate [22], exist to estimate $S(\omega)$ given a finite sequence of observations $\{x_k\}_{k=1}^K$. In our setting, due to the non-linearity of the model, these techniques cannot be directly applied. We therefore consider a discrete approximation to the PSD by assuming that the process $Z(\omega)$ defines a discrete-parameter harmonic process, i.e., it is constant over intervals of length $\frac{\pi}{N}$ for large enough N [22]. With this assumption, we can replace $Z(\omega)$ by a jump process in $[0, \pi)$ with jumps of $\frac{\pi}{N}(a_i + jb_i)$ at $\omega_i = \frac{i\pi}{N}$, where a_i and b_i are some random variables, for $i = 1, 2, \dots, N-1$, and $\frac{\pi}{N}$ is a normalization factor. Given that the process x_k is *real*, and invoking the symmetry $Z(\omega) = Z^*(-\omega)$, we can express the integral in Eq. (2.3) as:

$$x_k - \mu = \frac{2\pi}{N} \sum_{i=1}^{N-1} \Big(a_i \cos(\omega_i k) - b_i \sin(\omega_i k) \Big). \tag{2.4}$$

where $\omega_i := \frac{i\pi}{N}$. Using the property $E\{|dZ(\omega)|^2\} = S(\omega)d\omega$, the PSD at ω_i for $i = 1, 2, \dots, N-1$ can be expressed as:

$$S(\omega_i) = \frac{\pi^2}{N^2} \mathbb{E}\{a_i^2 + b_i^2\}. \tag{2.5}$$

Letting $\mathbf{x} = [x_1, x_2, \dots, x_K]^T \in \mathbb{R}^K$, $\mathbf{v} = [\frac{N}{2\pi}\mu, a_1, b_1, a_2, b_2, \dots, a_{N-1}, b_{N-1}]^T \in \mathbb{R}^{2N-1}$, and defining $\mathbf{A} \in \mathbb{R}^{K \times (2N-1)}$ as

$$\mathbf{A} := \tfrac{2\pi}{N} \begin{bmatrix} 1 & \cos(\frac{\pi}{N}) & -\sin(\frac{\pi}{N}) & \dots & \cos\left(\frac{(N-1)\pi}{N}\right) & -\sin\left(\frac{(N-1)\pi}{N}\right) \\ 1 & \cos(\frac{2\pi}{N}) & -\sin(\frac{2\pi}{N}) & \dots & \cos\left(\frac{2(N-1)\pi}{N}\right) & -\sin\left(\frac{2(N-1)\pi}{N}\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\frac{K\pi}{N}) & -\sin(\frac{K\pi}{N}) & \dots & \cos\left(\frac{K(N-1)\pi}{N}\right) & -\sin\left(\frac{K(N-1)\pi}{N}\right) \end{bmatrix} \tag{2.6}$$

we can express Eq. (2.4) as follows:

$$\mathbf{x} = \mathbf{A}\mathbf{v}. \tag{2.7}$$

It is worth noting that the matrix $\mathbf{A}$ resembles the DFT/DCT synthesis matrices; however, it is not in general full rank. $\mathbf{A}$ would have full column rank (resp. full row rank) if $K \geq 2N{-}1$ (resp. $K \leq 2N{-}1$). We further assume that the process $Z(\omega)$ is Gaussian, and hence the variables $v_i \sim \mathcal{N}(0, \sigma_i^2)$, for $i = 2, 3, \cdots, 2N-1$. Note that v_i's are independent due to the orthogonality of the increments of $Z(\omega)$. According to Eq. (2.5), we have $S(\omega_i) = \frac{\pi^2}{N^2}(\sigma_{2i}^2 + \sigma_{2i+1}^2)$, which corresponds to the discrete PSD approximation at $\omega_i = \frac{i\pi}{N}$, for $i = 1, \cdots, N-1$, with N controlling the degree of approximation. Since we are interested in the oscillatory behavior of the CIF, estimation of μ (i.e., the DC component) is not of particular importance. Nevertheless, in order to have a consistent prior on all the elements of $\mathbf{v}$, we assume an independent Gaussian prior on μ such that $v_1 \sim \mathcal{N}(0, \sigma_1^2)$.

2.2 Bayesian Estimation of the PSD

As a result of our formulation in Section 2.1, estimating the PSD of x_k is reduced to estimating the parameters $\boldsymbol{\theta} := [\sigma_1^2, \sigma_2^2, ..., \sigma_{2N-1}^2]^T$. Using a Bayesian formulation, we will perform the parameter estimation in a computationally efficient way. In addition, we can enforce the sparsity of the PSD by incorporating sparsity-promoting priors on $\boldsymbol{\theta}$. To this end, we use an exponential prior with parameter γ for the elements of $\boldsymbol{\theta}$ resulting in a log-prior $\log f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = (2N{-}1)\log\gamma - \gamma\sum_{i=1}^{2N-1}\sigma_i^2$. Note that the log-prior is akin to the ℓ_1-norm of $\boldsymbol{\theta}$ modulo constants, which is known to promote sparsity. Using the shorthand notation

$$\mathcal{D} := \left\{ n_k^{(\ell)} \right\}_{\ell=1,k=1}^{L,K}, \tag{2.8}$$

the maximum *a posteriori* (MAP) estimate of $\boldsymbol{\theta}$ is defined as:

$$\widehat{\boldsymbol{\theta}}_{\mathsf{MAP}} = \arg\max_{\boldsymbol{\theta}} \left(\log f_{\boldsymbol{\theta}|\mathcal{D}} \left(\boldsymbol{\theta}|\mathcal{D} \right) \right) = \arg\max_{\boldsymbol{\theta}} \Big(\log P(\mathcal{D}|\boldsymbol{\theta}) + \log f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \Big) \tag{2.9}$$

2.2.1 MAP Estimation via the Expectation-Maximization Algorithm

Expressing $P(\mathcal{D}|\boldsymbol{\theta})$ solely in terms of the data $\mathcal{D}$ results in an intractable function of $\boldsymbol{\theta}$. However, if the vector $\mathbf{v}$ was known, the log-likelihood of the *complete* data could be expressed as:

$$\begin{aligned} \log f(\mathcal{D}, \mathbf{v}|\boldsymbol{\theta}) =& \log P(\mathcal{D}|\mathbf{v}, \boldsymbol{\theta}) + \log f_{\mathbf{v}|\boldsymbol{\theta}}(\mathbf{v}|\boldsymbol{\theta}) \\ =& \sum_{k=1}^{K} \sum_{\ell=1}^{L} n_k^{(\ell)} (\mathbf{A}\mathbf{v})_k - \log\left(1 + \exp\left((\mathbf{A}\mathbf{v})_k\right)\right) \\ & - \sum_{i=1}^{2N-1} \left(\frac{v_i^2}{2\sigma_i^2} + \frac{1}{2} \log \sigma_i^2 \right) + \mathsf{cnst}. \end{aligned} \tag{2.10}$$

where $\mathsf{cnst.}$ stands for terms which are not functions of $\mathbf{v}$ or $\boldsymbol{\theta}$. We can thus use the Expectation-Maximization (EM) algorithm to calculate the MAP estimate in (2.9) [27].

The E Step: Suppose that at iteration r, we have an estimate of $\boldsymbol{\theta}$, denoted by $\widehat{\boldsymbol{\theta}}^{(r)} = \left[\sigma_1^{2(r)}, \sigma_2^{2(r)}, \cdots, \sigma_{2N-1}^{2}{}^{(r)} \right]^T$. Given that in the complete data $(\mathcal{D}, \mathbf{v})$, the vector $\mathbf{v}$ is unobserved, in the E step we calculate the function

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}^{(r)}\right) :=&\mathbb{E}_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}\big\{\log f(\mathcal{D},\mathbf{v}|\boldsymbol{\theta})\big\} + \log f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\
=&\mathbb{E}_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}\big\{\log f_{\mathbf{v}|\boldsymbol{\theta}}(\mathbf{v}|\boldsymbol{\theta})\big\} + \log f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + \mathsf{cnst.} \qquad (2.11)\\
=&\sum_{i=1}^{2N-1}\Big(-\frac{1}{2}\log\sigma_i^2 - \frac{1}{2\sigma_i^2}\mathbb{E}_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}\big\{v_i^2\big\} - \gamma\sigma_i^2\Big) + \mathsf{cnst.}
\end{aligned}
$$

where, similar to (2.10), we have used the conditional independence $P(\mathcal{D}|\mathbf{v},\boldsymbol{\theta}) = P(\mathcal{D}|\mathbf{v})$. In (2.11), the term $\mathsf{cnst.}$ represents all terms which are not functions of $\boldsymbol{\theta}$.

In order to compute the expectation in (2.11), the distribution of $\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}$ or its samples are required. However, calculating the distribution of $\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}$ involves computing intractable integrals, and sampling from $\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}$ by numerical methods such as the Metropolis-Hastings is not computationally efficient considering that it has to be carried out in every iteration. As a result, Monte Carlo methods are not computationally efficient when N is large.

As shown in [5,16], the density of $\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}$, which is proportional to the product of the Gaussian density $\mathbf{v}|\widehat{\boldsymbol{\theta}}^{(r)}$ and a Binomial $\mathcal{D}|\mathbf{v}$, can be well approximated by a multivariate Gaussian density $\mathcal{N}\big(\boldsymbol{\mu}_{\mathbf{v}}^{(r)}, \boldsymbol{\Sigma}_{\mathbf{v}}^{(r)}\big)$. Noting that the mean and mode of a multivariate Gaussian coincide, and that the Hessian of its natural logarithm is equal to $-\big(\boldsymbol{\Sigma}_{\mathbf{v}}^{(r)}\big)^{-1}$, we calculate the mode of $f_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}(v|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)})$ as the mean of the Gaussian approximation and the Hessian of $\log f_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}(v|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)})$ evaulated at the mode as $-\big(\boldsymbol{\Sigma}_{\mathbf{v}}^{(r)}\big)^{-1}$. For $\boldsymbol{\mu}_{\mathbf{v}}^{(r)}$, we have:

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{v}}^{(r)} &= \arg\max_{\mathbf{v}} \quad \log f_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}(v|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}) \\
&= \arg\max_{\mathbf{v}} \big(\log P(\mathcal{D}|\mathbf{v}) + \log f_{\mathbf{v}|\widehat{\boldsymbol{\theta}}^{(r)}}(\mathbf{v}|\widehat{\boldsymbol{\theta}}^{(r)})\big) \qquad (2.12) \\
&= \arg\max_{\mathbf{v}} \left(\sum_{k=1}^{K}\sum_{\ell=1}^{L} n_k^{(\ell)}(\mathbf{A}\mathbf{v})_k - \log\left(1+\exp\left((\mathbf{A}\mathbf{v})_k\right)\right) - \sum_{i=1}^{2N-1}\frac{v_i^2}{2\sigma_i^{2\,(r)}}\right).
\end{aligned}$$

The maximization problem in (2.12) is concave, and the Hessian is negative definite. Hence, we can use the Newton's method to efficiently compute $\boldsymbol{\mu}_{\mathbf{v}}^{(r)}$. The method is summarized in Algorithm 1. The stopping condition $\mathcal{S}_N$ can be either a convergence constraint or a limit on the number of iterations.

Algorithm 1 Newton's method for finding $\boldsymbol{\mu}_{\mathbf{v}}^{(r)}$

```
Inputs: ensemble average spiking data[a] $\bar{n}_k := \frac{1}{L}\sum_{\ell=1}^{L} n_k^{(\ell)}$, for $k = 1, 2, \cdots, K$,
    current parameter estimate $\widehat{\boldsymbol{\theta}}^{(r)}$, Newton's stopping condition $\mathcal{S}_N$.
Output: $\boldsymbol{\mu}_{\mathbf{v}}^{(r)}$.
$\mathbf{m}^{(0)} = \mathbf{0}$.
iteration number $i = 0$.
while $\neg\mathcal{S}_N$ do
    $i \leftarrow i + 1$
    $\mathbf{x} = \mathbf{A}\mathbf{m}^{(i-1)}$.
    $\lambda_k = \frac{1}{1+e^{-x_k}}$, for $1 \leq k \leq K$.
    $\mathbf{q} = \left[\frac{m_1^{(i-1)}}{\sigma_1^{2\,(r)}}, \ldots, \frac{m_{2N-1}^{(i-1)}}{\sigma_{2N-1}^{2\,(r)}}\right]^T$.
    calculate the gradient as $\mathbf{g} = L\mathbf{A}^T(\bar{\mathbf{n}} - \boldsymbol{\lambda}) - \mathbf{q}$
    $\mathbf{U} = \mathrm{diag}\left\{\frac{1}{\sigma_1^{2\,(r)}}, \ldots, \frac{1}{\sigma_{2N-1}^{2\,(r)}}\right\}$.
    $\mathbf{G} = \mathrm{diag}\left\{\frac{e^{-x_1}}{(1+e^{-x_1})^2}, \ldots, \frac{e^{-x_K}}{(1+e^{-x_K})^2}\right\}$.
    calculate the Hessian as $\mathbf{H} = -L\mathbf{A}^T\mathbf{G}\mathbf{A} - \mathbf{U}$.
    $\mathbf{m}^{(i)} = \mathbf{m}^{(i-1)} - \mathbf{H}^{-1}\mathbf{g}$.
end while
$\boldsymbol{\mu}_{\mathbf{v}}^{(r)} = \mathbf{m}^{(i)}$.
```

[a] $\bar{n}_k$ is often referred to as the Peristimulus Time Histogram (PSTH).

Letting $\mathbf{a}_i^T$ denote the ith row of the matrix $\mathbf{A}$, the inverse covariance can be computed as:

$$\left(\boldsymbol{\Sigma}_{\mathbf{v}}^{(r)}\right)^{-1} = L\mathbf{A}^T\mathbf{F}\mathbf{A} + \operatorname{diag}\left\{\frac{1}{\sigma_1^{2(r)}}, \cdots, \frac{1}{\sigma_{2N-1}^{2}{}^{(r)}}\right\} \tag{2.13}$$

where

$$\mathbf{F} = \operatorname{diag}\left\{\frac{e^{-\mathbf{a}_1^T\boldsymbol{\mu}_{\mathbf{v}}^{(r)}}}{(1+e^{-\mathbf{a}_1^T\boldsymbol{\mu}_{\mathbf{v}}^{(r)}})^2}, \cdots, \frac{e^{-\mathbf{a}_K^T\boldsymbol{\mu}_{\mathbf{v}}^{(r)}}}{(1+e^{-\mathbf{a}_K^T\boldsymbol{\mu}_{\mathbf{v}}^{(r)}})^2}\right\} \tag{2.14}$$

Going back to (2.11), using solutions of (2.12) and (2.13), we have $\mathbb{E}_{\mathbf{v}|\mathcal{D},\widehat{\boldsymbol{\theta}}^{(r)}}\left\{v_i^2\right\} = \left((\boldsymbol{\mu}_{\mathbf{v}}^{(r)})_i\right)^2 + \left(\boldsymbol{\Sigma}_{\mathbf{v}}^{(r)}\right)_{i,i}$, which we denote by $E_i^{(r)}$ for notational simplicity.

The M Step: In the M step, we maximize $Q(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}^{(r)})$ with respect to $\boldsymbol{\theta}$. The function $Q(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}^{(r)})$ is quasi-concave over the positive orthant with a unique maximizer. We have:

$$\frac{\partial}{\partial\sigma_i^2}Q(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}^{(r)}) = -\frac{1}{2\sigma_i^2} + \frac{E_i^{(r)}}{2\sigma_i^4} - \gamma = 0 \tag{2.15}$$

Noting that γ and $E_i^{(r)}$ are both positive, solving the quadratic equation $2\gamma\sigma_i^4 + \sigma_i^2 - E_i^{(r)} = 0$ for $1 \le i \le 2N-1$ in terms of σ_i^2's and picking the positive root gives the updated parameter vector as:

$$(\widehat{\boldsymbol{\theta}}^{(r+1)})_i = \sigma_i^{2(r+1)} = \frac{-1+\sqrt{1+8\gamma E_i^{(r)}}}{4\gamma}, \quad 1 \le i \le 2N-1 \tag{2.16}$$

It is worth noting that if no prior on σ_i^2's is used, the EM algorithm can be used similarly to calculate the Maximum Likelihood (ML) estimate of $\boldsymbol{\theta}$ given the data $\mathcal{D}$. In that case, the update rule of the EM algorithm is simply given by:

$$(\widehat{\boldsymbol{\theta}}^{(r+1)})_i = \sigma_i^{2(r+1)} = E_i^{(r)}, \quad 1 \le i \le 2N-1 \tag{2.17}$$

Algorithm 2 summarizes the MAP estimation of the PSD. The EM stopping condition $\mathcal{S}_{EM}$ again can be either a convergence condition or a limit on the number of iterations. A random vector with small positive elements can be considered as the initialization point $\boldsymbol{\theta}^{(0)}$. It is worth noting that the maximization problem of Eq. (2.9) is not concave or quasi-concave in general. Hence, the EM algorithm may converge to a local maximum, depending on its initialization. However, our numerical analysis suggests that initializing the EM algorithm with $\boldsymbol{\theta}^{(0)}$ taking small positive values, results in meaningful and interpretable estimates as shown in our simulations and real data analysis in Section 2.3.

Algorithm 2 MAP estimate of the PSD

```
Inputs: Ensemble spike observations $\mathcal{D} = \{n_k^{(\ell)}\}_{\ell=1,k=1}^{L,K}$, exponential prior hyper-
    parameter $\gamma$, EM stopping condition $\mathcal{S}_{EM}$, EM initialization $\boldsymbol{\theta}^{(0)}$, frequency
    spacing of the PSD estimate as the number of bins $N$ in $[0, \pi]$.
Output: $N-1$ uniform samples in $(0, \pi)$ of the PSD associated with the ensemble
    CIF.
Construct the matrix $\mathbf{A}$ as in Eq. (2.6).
Iteration number $r = 0$.
while $\neg\mathcal{S}_{EM}$ do
    Using Algorithm 1 with $\boldsymbol{\theta}^{(r)}$, solve the optimization problem of (2.12) to
    calculate the mean of the Gaussian approximation, i.e. $\boldsymbol{\mu}_{\mathbf{v}}^{(r+1)}$.
    Using $\boldsymbol{\theta}^{(r)}$ and $\boldsymbol{\mu}_{\mathbf{v}}^{(r+1)}$, calculate (2.13) as the covariance inverse of the Gaussian
    approximation, i.e. $\left(\boldsymbol{\Sigma}_{\mathbf{v}}^{(r+1)}\right)^{-1}$.
    Update $\boldsymbol{\theta}$ based on (2.16) to get $\widehat{\boldsymbol{\theta}}^{(r+1)}$
    $r \leftarrow r + 1$
end while
Using the last updated parameter vector $\widehat{\boldsymbol{\theta}}^{(r)}$, calculate the PSD estimates $\widehat{S}_i =$
$\frac{\pi^2}{N^2}({\sigma^2}_{2i}^{(r)} + {\sigma^2}_{2i+1}^{(r)})$ for $1 \leq i \leq N-1$.
```

Remark 2.1. It is worth noting that if there exists any prior information on the maximum frequency content of the data, this information can be incorporated into the model in order to reduce the computational cost. For instance, neural data is often sampled at rates much higher than the significant frequency content. Suppose f_{spc}

is the frequency spacing we require, and we know the maximum frequency content would not be larger than $f_{\max}$. Thus, The number of bins in $[0, \frac{f_s}{2})$ corresponding to this spacing is $N_{\text{spc}} = \lceil \frac{f_s}{2f_{\text{spc}}} \rceil$, and we want to focus on the first $N_{\max} = \lceil \frac{f_{\max}}{f_{\text{spc}}} \rceil$ bins. In this case, the matrix $\mathbf{A}$ in Eq. (2.6) can be reduced to an $M \times (2N_{\max} - 1)$ matrix rather than a $M \times (2N_{\text{spc}} - 1)$ matrix. This modification can greatly reduce the computational cost of the problem.

Remark 2.2. In general, the spectral resolution of a possibly infinite stationary signal depends on the number of acquired samples, i.e. the main lobe width of the sampling window. Similarly, the number of spiking samples K externally limits the frequency resolution in the spectrum estimate of the neural covariate. In order not to confuse this resolution with $\frac{\pi}{N}$, we keep referring to $\frac{\pi}{N}$ as the frequency spacing in the estimated spectrum rather than the spectrum resolution. As mentioned before, N represents the desired number of spectrum estimate samples in $[0, \pi)$ and controls the degree of approximation.

Remark 2.3. Note that the as long as $K \geq 2N_{\max} - 1$, the full column rank virtue of the matrix $\mathbf{A}$ guarantees stable estimates of the spectra from Eqs. (2.12) and (2.13). When $K \leq 2N_{\max} - 1$, the matrix will only have full row rank, which may result in instability of the Newton's algorithm. Although the ℓ_1-regularization in this case may mitigate the latter shortcoming (i.e., γ in Eq. (2.15)), we assume in what follows that the number of observations K satisfies $K \geq 2N_{\max} - 1$.

2.2.2 Hyper-Parameter Selection

We choose the optimal value of the hyper-parameter γ using cross-validation. We will use a two-fold cross-validation algorithm [28] to this end. We divide the ensemble into two groups, thereby partitioning the data into $\mathcal{D}_1$ and $\mathcal{D}_2$. In this case, the cross-validation criterion for each value of γ is the likelihood of $\mathcal{D}_1$ (res. $\mathcal{D}_2$) given the estimated parameter vector $\widehat{\boldsymbol{\theta}}$ using $\mathcal{D}_2$ (res. $\mathcal{D}_1$). Considering the generic data set $\mathcal{D}$, we have:

$$\begin{aligned} P\left(\mathcal{D}|\widehat{\boldsymbol{\theta}}\right) &= \underset{\mathbf{v}\in\mathbb{R}^{(2N-1)}}{\int\cdots\int} f(\mathcal{D},\mathbf{v}|\widehat{\boldsymbol{\theta}})dv_1dv_2\cdots dv_{2N-1} \qquad (2.18)\\ &= \underset{\mathbf{v}\in\mathbb{R}^{(2N-1)}}{\int\cdots\int} P(\mathcal{D}|\mathbf{v})f_{\mathbf{v}|\widehat{\boldsymbol{\theta}}}(\mathbf{v}|\widehat{\boldsymbol{\theta}})dv_1dv_2\cdots dv_{2N-1} = \mathbb{E}_{\mathbf{v}|\widehat{\boldsymbol{\theta}}}\Big\{P(\mathcal{D}|\mathbf{v})\Big\}. \end{aligned}$$

We also have:

$$P(\mathcal{D}|\mathbf{v}) = \prod_{k=1}^{K}\left(\frac{1}{1+e^{-\mathbf{a}_k^T\mathbf{v}}}\right)^{\sum\limits_{\ell=1}^{L} n_k^{(\ell)}}\left(1-\frac{1}{1+e^{-\mathbf{a}_k^T\mathbf{v}}}\right)^{\sum\limits_{\ell=1}^{L}(1-n_k^{(\ell)})}. \qquad (2.19)$$

Due to the independent Gaussian priors on the elements of $\mathbf{v}$ in our model, we have $\mathbf{v}|\widehat{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0},\text{diag}\{\widehat{\boldsymbol{\theta}}\})$. Thus, the expectation in (2.18) can be estimated in arbitrary precision using the Monte Carlo method [28] by drawing R samples $\mathbf{v}_1,\ldots,\mathbf{v}_R$ from $\mathcal{N}(\mathbf{0},\text{diag}\{\widehat{\boldsymbol{\theta}}\})$ and calculating the sample average of $P(\mathcal{D}|\mathbf{v})$ in (2.19), i.e., $P(\mathcal{D}|\widehat{\boldsymbol{\theta}}) \simeq \frac{1}{R}\sum_{i=1}^{R} P(\mathcal{D}|\mathbf{v}_r)$. Note that in order to ensure numerical stability, we compute $\log P(\mathcal{D}|\mathbf{v}_r)$, which from Eq. (2.19) takes an additive form over k. Algorithm 3 summarizes the steps of the cross-validation algorithm to determine the optimal value of the hyper-parameter γ_{opt} among a set of test values.

Algorithm 3 Two-fold cross-validation for optimizing the hyper-parameter γ

```
Inputs: Two subsets of data $\mathcal{D}_1$ and $\mathcal{D}_2$, and a set of candidate values of $\gamma$ given
    by $\Gamma$.
Output: Optimal value of the hyperparameter $\gamma_{\text{opt}}$.
1: for each test value of $\gamma$ do
2:     Estimate $\widehat{\boldsymbol{\theta}}_1$ using $\mathcal{D}_1$ from Algorithm 2.
3:     Draw $R$ samples $v_1, \ldots, v_R$ from $\mathcal{N}(\mathbf{0}, \text{diag}\{\widehat{\boldsymbol{\theta}}_1\})$.
4:     Estimate $\mathcal{L}_{2|1} := P(\mathcal{D}_2|\widehat{\boldsymbol{\theta}}_1)$ using Monte Carlo sampling from (2.19).
5:     Repeat steps 2 to 4 interchanging the roles of $\mathcal{D}_1$ and $\mathcal{D}_2$ to calculate $\mathcal{L}_{1|2}$.
6:     $\mathcal{L}(\gamma) = \frac{1}{2}(\mathcal{L}_{2|1} + \mathcal{L}_{1|2})$.
7: end for
8: $\gamma_{\text{opt}} = \arg\max_{\gamma \in \Gamma} \mathcal{L}(\gamma)$.
```

2.2.3 Constructing Confidence Intervals

It is possible to construct confidence intervals for the estimated PSD values $\{\widehat{S}_1, \cdots, \widehat{S}_{N-1}\}$ obtained from Algorithm 2 by sampling from the density $f_{\boldsymbol{\theta}|\mathcal{D}}(\widehat{\boldsymbol{\theta}}|\mathcal{D})$. We have:

$$f_{\widehat{\boldsymbol{\theta}}|\mathcal{D}}(\widehat{\boldsymbol{\theta}}|\mathcal{D}) \propto P(\mathcal{D}|\widehat{\boldsymbol{\theta}}) f_{\boldsymbol{\theta}}(\widehat{\boldsymbol{\theta}}) \propto \underbrace{\mathbb{E}_{\mathbf{v}|\widehat{\boldsymbol{\theta}}}\Big\{P(\mathcal{D}|\mathbf{v})\Big\} e^{-\gamma \sum\limits_{i=1}^{2N-1} \sigma_i^2}}_{g(\widehat{\boldsymbol{\theta}}, \mathcal{D}, \gamma)} \tag{2.20}$$

We can therefore use the Metropolis-Hastings algorithm [28] to sample from $f_{\boldsymbol{\theta}|\mathcal{D}}(\widehat{\boldsymbol{\theta}}|\mathcal{D})$. The expectation term in $g(\widehat{\boldsymbol{\theta}}, \mathcal{D}, \gamma)$ can be estimated using the Monte Carlo sampling procedure explained in Section 2.2.2. Algorithm 4 summarizes the Metropolis-Hasting algorithm for our sampling purpose. For simplicity, we have considered a Gaussian proposal density $q(\mathbf{u}|\mathbf{w})$, i.e. $\mathbf{u}|\mathbf{w} \backsim \mathcal{N}(\mathbf{u}, \boldsymbol{\Sigma}_q)$, where $\boldsymbol{\Sigma}_q$ is a diagonal matrix in $\mathbb{R}^{(2N-1)\times(2N-1)}$ with diagonal elements proportional to the estimated parameter vector $\widehat{\boldsymbol{\theta}}$. It is worth noting that $g(\widehat{\boldsymbol{\theta}}, \mathcal{D}, \gamma)$ is zero for any vector $\widehat{\boldsymbol{\theta}}$ which is not element-wise non-negative. Thus, if the normally distributed candidate $\mathbf{z}$ in line 3 of Algorithm 4 has negative components, it would be discarded.

Algorithm 4 Constructing confidence intervals for $\{\widehat{S}_i\}_{i=1}^{N-1}$

```
Inputs: Neural spiking data $\mathcal{D}$, cross-validated hyperparameter $\gamma_{\text{opt}}$, MAP estimate of $\boldsymbol{\theta}$ ($\widehat{\boldsymbol{\theta}}_{\mathsf{MAP}}$), number of samples $M$, symmetric proposal density function $q(.|.)$.
Output: confidence intervals for $\{\widehat{S}_i\}_{i=1}^{N-1}$.
Initialize $\boldsymbol{\vartheta}^{(0)} = \widehat{\boldsymbol{\theta}}_{\mathsf{MAP}}$.
while $m \leq M$ do
    Generate a candidate $\mathbf{z}$ for the next sample by sampling from the density $q(\mathbf{z}|\boldsymbol{\vartheta}^{(m-1)})$.
    Calculate the acceptance ratio $\alpha := \frac{g(\mathbf{z},\mathcal{D},\gamma_{opt})}{g(\boldsymbol{\vartheta}^{(m-1)},\mathcal{D},\gamma_{opt})}$.
    If $\alpha \geq 1$, accept $\mathbf{z}$ as the next sample; otherwise, accept $\mathbf{z}$ as the next sample with probability $\alpha$.
    If $z$ is accepted as the next sample set $\boldsymbol{\vartheta}^{(m)} = \mathbf{z}$; otherwise set $\boldsymbol{\vartheta}^{(m)} = \boldsymbol{\vartheta}^{(m-1)}$.
    $m \leftarrow m + 1$
end while
Transform each sample $\boldsymbol{\vartheta}^{(m)}, 1 \leq m \leq M$, into a sample for $\widehat{S}_i$ as $\widehat{S}_i^{(m)} = \frac{\pi^2}{N^2}({\sigma^2}_{2i}^{(m)} + {\sigma^2}_{2i+1}^{(m)})$ for $1 \leq i \leq N-1$.
Construct confidence intervals at a level $1-\nu$ for $\{\widehat{S}_i\}_{i=1}^{N-1}$ using the samples $\{\widehat{S}_i^{(m)}\}_{m=1,i=1}^{M,N-1}$.
```

2.3 Application to Simulated and Real Data

In this section, we first demonstrate the performance of our method in two simulated settings. The two settings correspond to CIFs from noisy two-tone line spectra and an autoregressive process, respectively. We compare the performance of our method with three existing techniques: 1) calculating the periodogram of the spiking data of each neuron and averaging over the periodograms across neurons, which we refer to as PER-PSD [29]; 2) smoothing the ensemble average spiking (PSTH) and computing the PSD of the resulting smoothed PSTH, which we refer to as the PSTH-PSD; 3) Using a state-space model to estimate the CIF, followed by computing the PSD of the estimated CIF [16], which we will refer to as SS-PSD.

Finally, we apply our method to ECoG data from a human subject under general anesthesia and compare the extracted PSD from the spiking data to that of the LFP.

It is worth noting that the class of spectra which is identifiable from neuronal spiking data, is generally limited by the spiking rate in the PSTH. In a way, the spiking rate represents the amount of information available for inference procedures such as spectral estimation. The higher the PSTH spiking rate is, the larger and more complex the class of identifiable spectra would be. In order to conduct simulation studies and perform comparisons with existing methods in a setting akin to real neuronal data, we have limited the simulation setting to PSTH spiking rates of %5–%10 and sparse spectra which can potentially be identified under these low spiking rates.

2.3.1 Spike Trains Driven by a Noisy Dual-Tone Signal

Consider the dual-tone signal

$$x(t) = 1.48\cos(2\pi f_0 t) + 0.685\cos(2\pi f_1 t) + 0.17n(t) - 5.7 \tag{2.21}$$

with $f_0 = 1$ Hz, $f_1 = 10$ Hz, and $n(t)$ representing a zero-mean white Gaussian noise with unit variance. When sampled at $f_s = 300$ Hz, the discretized data forms x_k. The bias term of -5.7 is chosen to make sure that the resulting spiking rate is low enough and consistent with real-world neuronal spiking rates. The tones at f_0 and f_1 are chosen as a model of neuronal spiking modulated by slow and alpha oscillations, respectively. We consider $K = 1000$ samples of x_k and simulated the spiking data for $L = 10$ neurons based on our model in (2.2). The signal x_k and the raster plot of the ensemble are shown in Figure 2.1–(a) and 2.1–(b), respectively. The average spiking rate of the ensemble from the PSTH is 0.056.

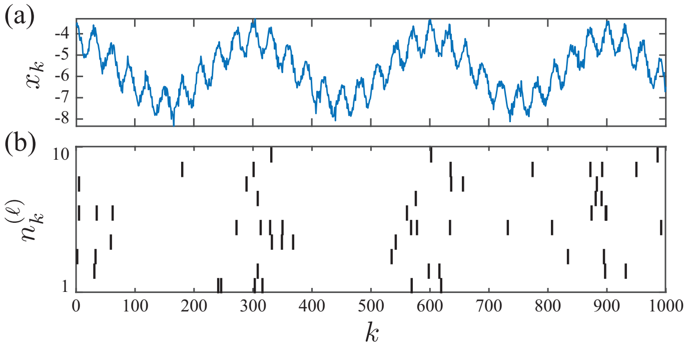

Figure 2.1: (a) Dual-tone signal $x_{1:K}$ (b) Raster plot of the ensemble.

Figure 2.2 shows the results obtained by our proposed method as well as the PER-PSD, PSTH-PSD and SS-PSD methods. Figure 2.2–(a) shows the PSTH smoothed via two Gaussian kernels: a narrow kernel (green trace) and a wide kernel (dotted red trace). Figure 2.2–(b) shows the normalized multitaper estimate [22,30, 31] of the PSD corresponding to the smoothed PSTH values shown in panel 2.2–(a) as well as the PER-PSD estimate. The multitaper method is arguably the most reliable nonparametric spectral estimation technique, as it addresses the estimation bias and variance trade-off in an optimal fashion [30]. The spectral resolution of the multitaper method is chosen as 0.125Hz. The means of all spiking signals in PER-PSD method, smoothed $\bar{n}_k$s in PSTH-PSD method, and $\widehat{x}_{k|K}$s in SS-PSD method are subtracted prior to calculating the periodogram or multitaper estimate to make sure the significant oscillatory components would not get dominated by the DC component. Also, all of the PSD estimates are normalized for comparison.

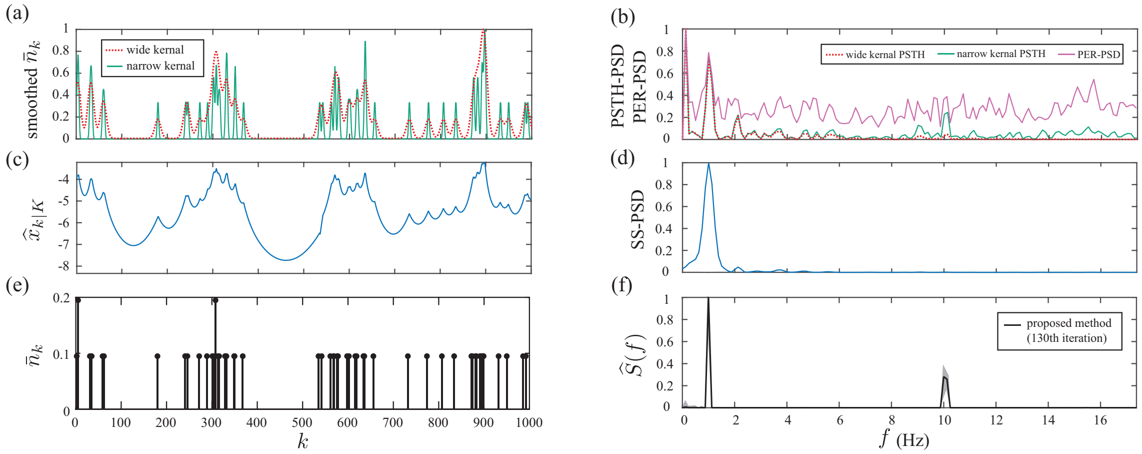

Figure 2.2: Noisy dual-tone CIF model for a neuronal ensemble: (a) Normalized smoothed PSTH using Gaussian kernels with small and large variances (b) Normalized PER-PSD estimate and normalized multitaper estimate of the PSD corresponding to the smoothed PSTHs (c) Estimate of x_k using state-space smoothing (d) Normalized multitaper estimate of the PSD of $\widehat{x}_{k|K}$ (e) Raw PSTH of the data $\bar{n}_k$ with 0.056 spiking rate (f) Normalized PSD estimate using the proposed method after 130 EM iterations together with %95 confidence intervals.

The PER-PSD method considers each spiking signal as samples of a stationary signal, and does not make use of the ensemble average (PSTH). Since periodogram is not a consistent estimator of the PSD and needs further smoothing, the average of the resulting periodograms across the realizations is calculated [29]. Hence, the PER-PSD method can be viewed as the average of the periodogram PSD estimates obtained from individual neurons. Figure 2.2–(b) (purple trace) shows the PER-PSD estimate. Although a significant peak is retrieved at 1Hz, the 10Hz component is not recovered due to the high variability of the estimate.

The estimate corresponding to the narrow smoothing kernel in PSTH-PSD

(Figure 2.2–(b), green trace) detects the two peaks at 1 Hz and 10 Hz, but has a high variability in higher frequencies. In addition, two spurious peaks at 2 Hz and 9 Hz are detected in the PSD. On the other hand, the estimate corresponding to the wide smoothing kernel (Figure 2.2–(b), dotted red trace) has a smaller variability but misses the 10 Hz component of the data. These results show the high sensitivity of the PSTH-PSD approach to the choice of the smoothing kernel. Comparing to PER-PSD, the PSTH-PSD method results in estimates with lower variability as it forms PSD estimates by employing the more informative ensemble average signal (PSTH) rather than the spiking data of the individual neurons.

Figure 2.2–(c), shows the estimate of x_k using state-space smoothing [16]. The state-space framework corresponding to our model is given by:

$$\begin{cases} x_k = x_{k-1} + \epsilon_k, \quad 1 \le k \le K \\ \epsilon_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \nu^2), \quad 1 \le k \le K \\ \lambda_k = \frac{1}{1+e^{-x_k}}, \\ n_k^{(\ell)} \sim \text{Bernoulli}(\lambda_k), \quad 1 \le k \le K, 1 \le \ell \le L \end{cases} \tag{2.22}$$

Using a forward/backward filtering, this method computes the MAP estimate of x_k given all the data, denoted by $\widehat{x}_{k|K}$, which is plotted in Figure 2.2–(c). The parameter ν^2 is estimated via the EM algorithm. Similar Gaussian density approximations to (2.12) and (2.13) have been used in [16] specially in the filtering and smoothing parts. As observed from Figure 2.2–(c), the estimate $\widehat{x}_{k|K}$ correlates with the smoothed PSTH estimate via a wide kernel (Figure 2.2–(a), dotted red trace). The normalized multitaper estimate of the PSD of $\widehat{x}_{k|K}$, namely the SS-PSD estimate, is shown in Figure 2.2–(d). Similar to the PSTH-PSD estimate using a wide Gaussian kernel, the 10 Hz cannot be recovered using this method.

The raw PSTH of the data $\bar{n}_k = \frac{1}{L}\sum_{\ell=1}^{L} n_k^{(\ell)}$, $1 \leq k \leq K$ is shown in Figure 2.2–(e), followed by our estimate of the PSD in Figure 2.2–(f). We have chosen $N = 1200$ corresponding to a frequency binning of 0.125 Hz for $f_s = 300$ Hz, which is comparable to the design resolution of the multitaper method used for the PSTH-PSD and SS-PSD methods. We have used the first 140 bins (0.125 Hz to 17.375 Hz) in constructing the matrix A, in order to reduce the computational complexity ($N_{\max} = 140$). Furthermore, the 95% confidence intervals for the identified oscillatory components are calculated using Algorithm 4 with $M = 1000$ samples and shown in Figure 2.2–(f) (gray hulls). The upper confidence bound at $f = 10$ Hz is at ≈ 1.4 and is truncated in the graph for graphical convenience. The cross-validated value for γ using Algorithm 3 is 10^{-4}. Clearly, both of the tones at 1 Hz and 10 Hz are recovered (unlike the PSTH-PSD with a wide kernel and the SS-PSD), while the irrelevant frequencies are significantly suppressed (unlike the PER-PSD method and the PSTH-PSD with a narrow kernel). Note that we have used 130 EM iterations to obtain the estimate. Figure 2.3 shows the estimated PSD vs. EM iterations. The dominant frequency of 1 Hz is detected at around iteration 30, and by continuing the EM iterations the component at 10 Hz is eventually discovered at around iteration 100. A MATLAB implementation of our algorithm, as well as the existing ones, producing Figure 2.2 is archived on the open source repository GitHub and made publicly available [32].

The foregoing simulation demonstrated the superior performance of our method

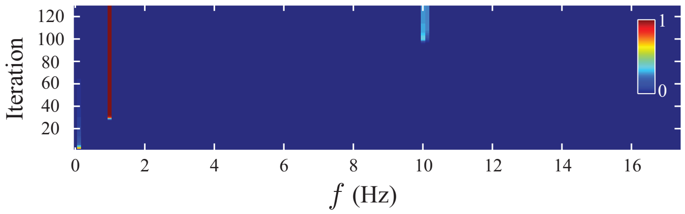

Figure 2.3: PSD vs. EM iterations corresponding to Figure 2.2–(f).

in application to ensemble neuronal activity. An intriguing comparison setting is to assess the performance of our method, as well as the existing ones, when applied to spiking data from a single neuron ($L = 1$). As mentioned earlier, we consider spiking data from a single neuron with sufficiently high spiking rate in order for the PSD estimation methods to have meaningful results. To this end, for the single neuron simulation, we change the bias term of -5.7 in Eq. (2.21) for the noisy dual-tone neural covariate to -3.7. This results in the average spiking rate of 0.05 for the single neuron spike train shown in Figure 2.4–(e). All the other parameters, including those of the dual-tone neural covariates, are the same as those in the foregoing simulation setting.

Figure 2.4 shows the estimated PSDs using the different methods applied to the spike train from a single neuron. Since the output of the narrow kernel PSTH-PSD method is nearly identical to that of the PER-PSD method in the case of a single neuron, we have only shown the PSD estimate of the PSTH-PSD method in Figure 2.4–(b). Similar to the foregoing simulation results using an ensemble of $L = 10$ neurons, we observe that the narrow kernel PSTH-PSD method identifies the two frequency peaks at 1Hz and 10Hz. However, the estimate also contains significant

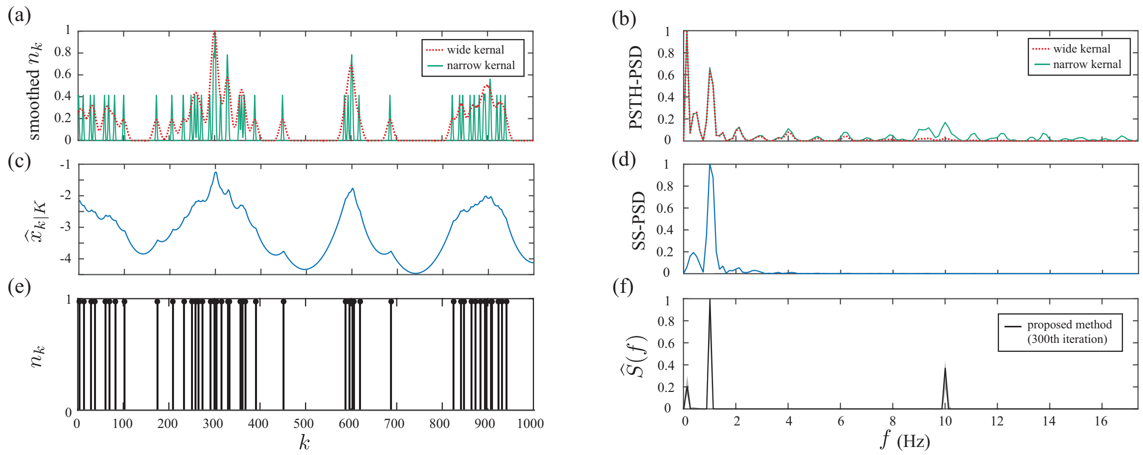

Figure 2.4: Noisy dual-tone CIF model for a single neuron: (a) Normalized smoothed spiking signal using Gaussian kernels with small and large variances (b) Normalized PSD estimates corresponding to the smoothed spiking signal (c) Estimate of x_k using state-space smoothing (d) Normalized multitaper estimate of the PSD of $\widehat{x}_{k|K}$ (e) Single neuron spiking signal n_k with 0.05 spiking rate (f) Normalized PSD estimate using the proposed method after 300 EM iterations together with %95 confidence intervals.

redundant peaks at 2Hz, 4Hz, and around 9Hz, while having a high variability at higher frequencies. Furthermore, both the wide kernel PSTH-PSD method and the SS-PSD method smooth the spiking data to the degree that results in missing the 10Hz peak in the PSD. In contrast, as shown in Figure 2.4–(f) and Figure 2.5, our proposed method recovers the two peaks and only contains a minor redundant low-frequency component in the PSD estimate. Note that since only one spiking realization is employed by our algorithm, it takes more EM iterations to decode the smaller peak at 10 Hz in this case, as compared to the foregoing simulation ($\sim$ 200 EM iterations in Figure 2.5 vs. $\sim$ 100 EM iterations in Figure 2.3).

In summary, these two simulation studies demonstrate the superior performance of our algorithm as compared to several existing techniques. In addition,

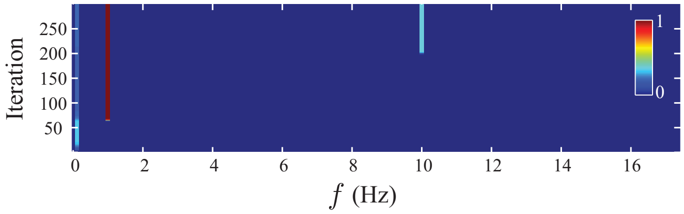

Figure 2.5: PSD vs. EM iterations corresponding to Figure 2.4–(f).

they highlight the difference of considering the ensemble spiking data vs. single-neuron spiking data for PSD estimation. If the PSTH corresponding to an ensemble of low spiking neurons is rich enough to identify a specific spectral structure, in order to get comparable results using data from a single neuron, the spiking rate must be chosen high enough to account for the lack of multiple realizations. In other words, an ensemble of low spiking neurons can provide much more information than considering each of them in isolation. This observation explains the performance gap between the PER-PSD and PSTH-PSD estimates shown in Figure 2.2–(b); the PER-PSD method forms periodogram estimates using single neuron spiking data rather than the PSTH, and hence exhibits inferior performance in comparison to the PSTH-PSD methods.

2.3.2 Spike Trains Driven by an AR(6) Process

In the second set of simulations, we examine a more complex scenario where the driving signal x_k is generated from a 6th order autoregressive (AR) process. Figure 2.6–(a) shows the PSD corresponding to the AR process with third-order poles at $\omega_1 = \frac{\pi}{20} = 0.1571$ rad and $\omega_2 = \frac{\pi}{5} = 0.6283$ rad with magnitudes of 0.997 and 0.999, respectively. The sample realization of this process of length $K = 500$ as well as the raster plot of the spike trains for $L = 10$ neurons are depicted in Figures 2.6–(b) and –(c), respectively. Similar to the previous simulation, a significant negative mean is added to the AR signal to make the PSTH spiking rate close to real neuronal spiking rates. The average spiking rate of the PSTH corresponding to the ensemble in Figure 2.6–(c) is 0.058.

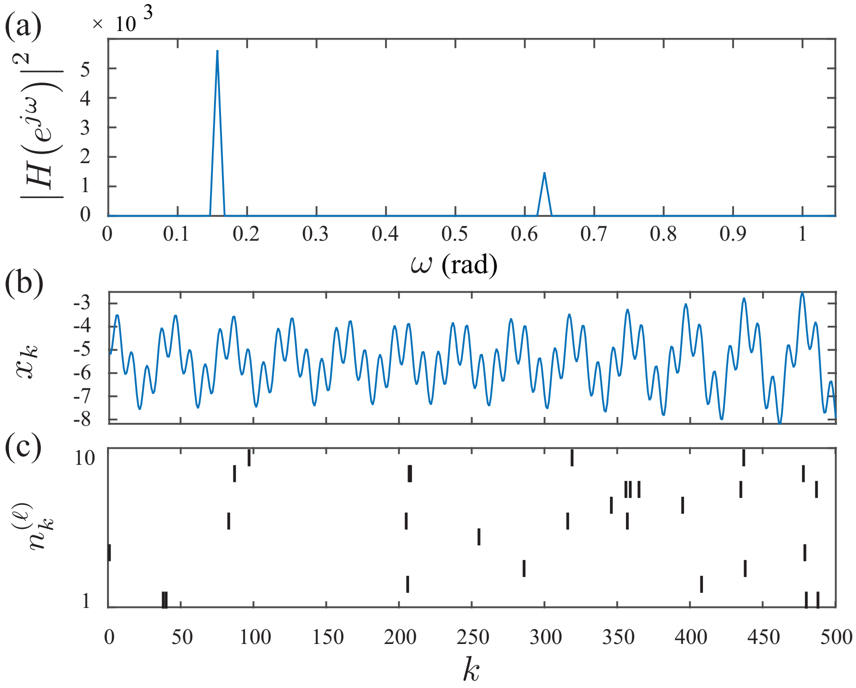

Figure 2.6: (a) PSD of the dual peak AR process (b) Sample realization of the AR process (c) Raster plot of the ensemble.

Figure 2.7 shows the estimated PSDs using the different methods. The spacing of our estimate is 0.105 rad corresponding to $N = 300$ bins in our model, out of which 100 bins have been considered to cover 0.0105 rad to 1.0395 rad frequency range ($N_{\max} = 100$). Similar to the previous example, the PER-PSD estimate in

Figure 2.7–(b) (purple trace) is so noisy that the two peaks at 0.1571 rad and 0.6283 rad are comparable to noisy retrieved components. Also in the PSTH-PSD method, we observe that the multitaper estimate corresponding to the narrow kernel in 2.7–(b) tends to have a high variability and detects undesired peaks around 0.4 rad and 0.8 rad, comparable in magnitude to the correctly identified peaks at 0.1571 rad and 0.6283 rad. While reducing the variability at higher frequencies, the wide smoothing Gaussian kernel has resulted in dismissing the peak at 0.6283 rad. Thus, the inevitable effect of tuning the width of the smoothing kernel persists in this example as well. Figures 2.7–(c) and 2.7–(d) show the results of the SS-PSD method in time and frequency domains, respectively. Again, we observe a correlation between the estimated $\widehat{x}_{k|K}$ in Figure 2.7–(c) and the smoothed $\bar{n}_k$ using a wide Gaussian kernel in Figure 2.7–(a). However, the SS-PSD method similarly dismisses the peak at 0.6283 rad. In addition, the recovered peak at 0.1571 rad is dominated by other falsely recovered low frequency components due to the temporal smoothing nature of the estimated $\widehat{x}_{k|K}$. Figures 2.7–(e) and 2.7–(f) respectively show the PSTH and the output of our method for 100 EM iterations, with the cross-validated value of 0.045 for γ. The convergence of the EM algorithm follows a similar pattern to that of the preceding section and is depicted in Figure 2.8. Similar to the previous simulation setting, $M = 1000$ samples are used for constructing 95% confidence intervals (grey hulls). The upper confidence bound at $\omega = 0.1571$ rad is at ≈ 1.2 and is truncated in the graph for graphical convenience. As observed in Figure 2.7–(f), the two peaks are perfectly recovered while the undesired frequency components are nearly estimated zero.

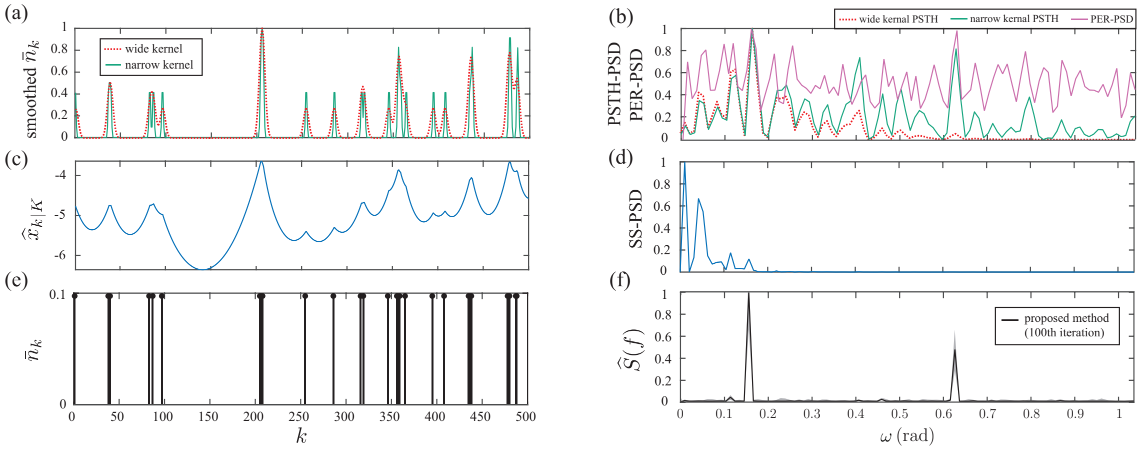

Figure 2.7: AR(6) generated CIF model for a neuronal ensemble: (a) Normalized smoothed PSTH using Gaussian kernels with small and large variances (b) Normalized PER-PSD estimate and normalized multitaper estimate of the PSD corresponding to the smoothed PSTHs (c) Estimate of x_k using state-space smoothing (d) Normalized multitaper estimate of the PSD of $\widehat{x}_{k|K}$ (e) Raw PSTH of the data $\bar{n}_k$ with 0.058 spiking rate (f) Normalized PSD estimate using the proposed method after 100 EM iterations together with %95 confidence intervals.

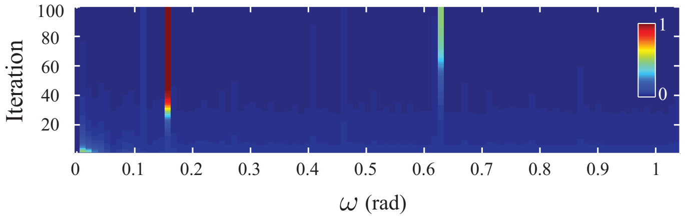

Figure 2.8: PSD vs. EM iterations corresponding to Figure 2.7–(f).

2.3.3 Application to Neuronal Spiking Data from Anesthesia

Finally, we apply our proposed algorithm on multi-unit recordings from a human subject under Propofol-induced general anesthesia (data from [33]). The data set includes the spiking activity of 41 neurons as well as the LFP recorded from a patient undergoing intra-cranial monitoring for surgical treatment of epilepsy using a multichannel micro-electrode array implanted in temporal cortex [33]. Recordings were conducted during the administration of Propofol for induction of anesthesia. The experimental protocol under which the data was collected is extensively explained in [33]. Given that the the original multi unit recordings are oversampled at a rate of 1 KHz, to reduce computational complexity, the spike recordings were downsampled by the factor of 40, and the sampling rate of the LFP signal was reduced from the original 250 Hz to 25 Hz. A time frame of 50 s is considered containing $K = 1250$ samples of the downsampled multi-unit recordings and the LFP signal. In our analysis, we have considered $L = 27$ neurons with at least two spikes in the 50 s time frame. Figure 2.9 shows the raster plot of the neuronal ensemble. The average spiking rate of the population from the PSTH is given by 0.1064.

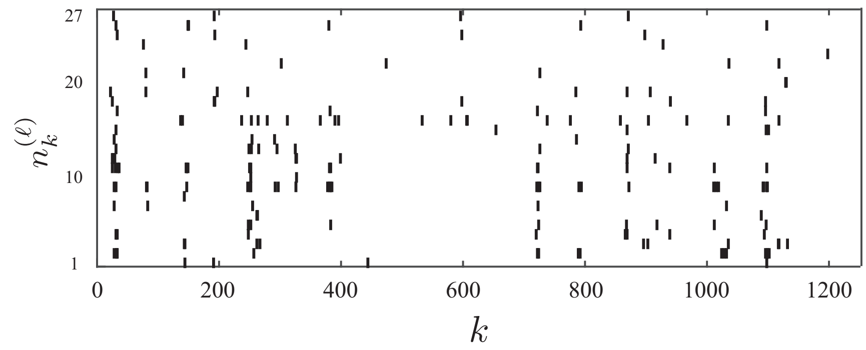

Figure 2.9: Raster plot of the neuronal ensemble corresponding to multi-unit recordings from a human subject under Propofol-induced general anesthesia.

Figure 2.10 shows the results of the different PSD estimation techniques. For our method, we have chosen a spacing of 0.02 Hz which corresponds to $N = 625$ frequency bins in our model, considering the reduced sampling rate of 25 Hz. Given that the relevant frequencies modulating neuronal spiking under anesthesia pertain to slow oscillations [33], we have considered the first 100 frequency bins in our method covering 0.02 Hz to 2 Hz ($N_{\max} = 100$). Figures 2.10–(a) and 2.10–(b) respectively show the smoothed PSTH with the two Gaussian kernels and their corresponding PSD estimates as well as the PER-PSD estimate. As was the case in our simulation studies, the PER-PSD estimate is noisy everywhere. Also, the PSTH-PSD estimate corresponding to the narrow kernel contains considerable variability in high frequencies. In contrast, the PSTH-PSD estimate using the wide smoothing kernel significantly suppresses the PSD components beyond 0.4 Hz. Figures 2.10–(c) and –(d) show the estimates of $x_{k|K}$ and the PSD using the SS-PSD method. Similar to the preceding simulation studies, low frequency components dominate the PSD estimate due to the heavy time-domain smoothing in estimating $x_{k|K}$.

The PSTH and the output of our method after 100 EM iterations ensuring convergence are shown in Figures 2.10–(e) and 2.10–(f), respectively. The EM convergence vs. iteration is shown in Figure 2.11. The cross-validated value for γ is 0.075, and $M = 1000$ samples are used in Algorithm 4 to construct 95% confidence intervals (grey hulls). The upper confidence bound at $f = 0.42$ Hz is at ≈ 1.35 and is truncated in the graph for graphical convenience. Figure 2.10–(g) and 2.10–(h) show the LFP and its multitaper PSD estimate respectively. The PSD of the LFP signal shows a dominant peak around 0.42 Hz, with a few others extending to 0.8 Hz. Strikingly, the PSD obtained by our method is the most similar to the PSD of the LFP: the PSTH-PSD fails to suppress the high frequency variability (narrow kernel) or dismisses the PSD peaks beyond 0.42 Hz (wide kernel). The SS-PSD

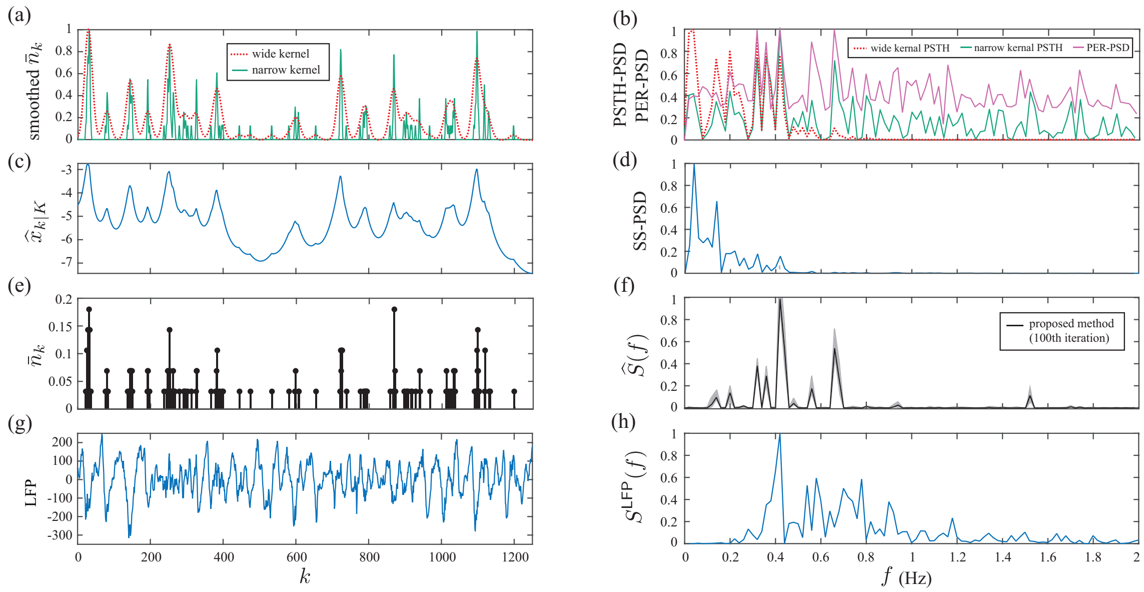

Figure 2.10: Neuronal spiking data from anesthesia: (a) Normalized smoothed PSTH using Gaussian kernels with small and large variances (b) Normalized PER-PSD estimate and normalized multitaper estimate of the PSD corresponding to the smoothed PSTHs (c) Estimate of x_k using state-space smoothing (d) Normalized multitaper estimate of the PSD of $\widehat{x}_{k|K}$ (e) Raw PSTH of the data $\bar{n}_k$ with 0.1064 spiking rate (f) Normalized PSD estimate using the proposed method after 100 EM iterations together with %95 confidence intervals (g) Recorded LFP signal (h) Normalized multitaper estimate of the PSD corresponding to the recorded LFP signal.

method recovers dominant low frequency component which do not exist in the PSD of the LFP signal. This result corroborates the findings of [33] that the neuronal spiking under general anesthesia is highly phase-locked to the LFP signal, and hence the LFP can be considered as a salient neural covariate driving the spiking of the nearby cortical neuronal ensemble.

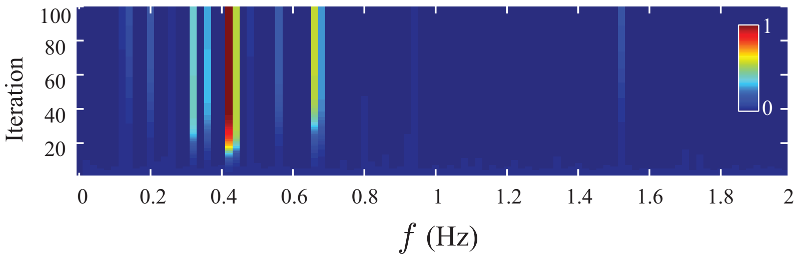

Figure 2.11: PSD vs. EM iterations corresponding to Figure 2.10–(f).

2.4 Discussion

The preceding section demonstrated the superior performance of our algorithm on simulated data as well as real data recordings. In order to explain this performance gap as well as to characterize the computational cost of our algorithm, two discussion points are in order.

2.4.1 Pursuit Domain Comparisons of the PSD Estimators

The significant performance gain of our proposed PSD estimation framework over methods such as the PSTH-PSD or PER-PSD mainly stems from the difference in the underlying pursuit domains. The PSTH-PSD and PER-PSD methods generate the spectral estimates by forming a second-order combination of the data from a finite collection of binary sets, i.e., Fourier transforms of the PSTH $\bar{n}_k = \frac{1}{L}\Sigma_{l=1}^{L} n_k^{(l)}$ or the spiking data of each neuron $n_k^{(l)}$, in which $n_k^{(l)} \in \{0, 1\}$. Given that the PSTH signal takes values in the set $\{0, \frac{1}{L}, \cdots, \frac{L-1}{L}, 1\}$ and the spiking of each neuron is a binary variable, the pursuit domains of the PSTH-PSD and PER-PSD algorithms are limited to small and finite subsets of $\mathbb{R}$. Perhaps the poor performance of the PER-PSD compared to the PSTH-PSD method is due to the fact that the former generates estimates using a second-order function of variables in $\{0, 1\}$, while the latter does so using the richer set $\{0, \frac{1}{L}, \cdots, \frac{L-1}{L}, 1\}$.

In contrast, our MAP estimator employs the same observations from the small subset $\{0, \frac{1}{L}, \cdots, \frac{L-1}{L}, 1\}$, but performs inference of the latent variables $\mathbf{x}$, $\mathbf{v}$, and $\boldsymbol{\theta}$ directly over the richer set $\mathbb{R}$. To this end, the MAP estimates are obtained by solving an optimization problem seeking a spectral estimate with elements in $\mathbb{R}$ that is consistent with the observed data and sparse priors in the Bayesian sense. Therefore, by searching over a much richer set, the MAP-based PSD estimator outperforms methods such as PSTH-PSD or PER-PSD. The SS-PSD method also searches for the latent variable $\mathbf{x}$ in $\mathbb{R}$, but due to enforcing smoothness of x_k in the time domain, generates spectral estimates which undergo distortion in the spectral domain.

2.4.2 Computational Comparisons of the PSD Estimators

The memory requirement of our approach is $\mathcal{O}(KN_{\max})$, as we need to store the matrix $\mathbf{A} \in \mathbb{R}^{K \times (2N_{\max}-1)}$. For each EM iteration, we have a concave optimization problem in Eq. (2.12) which includes a logistic log-likelihood as its objective function plus a quadratic term. As Newton's method is widely used for logistic as well as approximately quadratic regression problems, we have chosen to use it as our main optimization algorithm. To achieve quadratic convergence, the Newton's method requires the calculation and inversion of the Hessian of the objective function. Calculation of the Hessian in line 11 of Algorithm 1 requires $\mathcal{O}(KN_{\max}^2)$ operations as $\mathbf{G}$ is diagonal and its inversion in line 12 has a computational cost of the order $\mathcal{O}(N_{\max}^3)$. To reduce the computational cost, we can also use quasi-Newton methods such as BFGS. While enjoying a super-linear convergence, these methods require $\mathcal{O}(KN_{\max})$ operations to calculate the gradient and $\mathcal{O}(N_{\max}^2)$ operations for the Hessian approximation [34]. For each EM iteration, the calculation of $E_i^{(r)}$ in Eq. (2.16) for $i = 1, \cdots, 2N_{\max}-1$, requires a matrix inversion with a cost of $\mathcal{O}(N_{\max}^3)$ operations. However, since we only need the diagonal elements of the inverse, this complexity can be reduced by methods in [35], in case $N_{\max}$ is large. In summary, the computational complexity of Algorithm 2 is $\mathcal{O}(KN_{\max}^2)$, if Newton's method is used, under the assumption of $K \geq 2N_{\max} - 1$. The PSTH-PSD, SS-PSD, and PER-PSD algorithms, however, have a complexity of $\mathcal{O}(K \log K)$, thanks to the underlying FFT procedure, in computing K samples of the PSD. There are other steps in our method, such as the cross-validation for the selection of the sparsity hyperparameter γ (Algorithm 3), and constructing confidence intervals (Algorithm 4). However, these steps involve multiple runs of Algorithm 2, and are common in MAP-based inference algorithms such as the SS-PSD algorithm. In light of the preceding comparison, the performance gain of our proposed algorithm comes with an increase of $\mathcal{O}\left(\frac{N_{\max}^2}{\log K}\right)$ in computational complexity.

2.5 Concluding Remarks

In this chapter, we considered the problem of computing the power spectral density of the neural covariates underlying spiking data. Existing methods first compute an estimate of the ensemble PSTH or CIF through a temporal smoothing procedure. Then, the PSD estimates of the smoothed PSTH or CIF are computed as the spectral representation of the data. This two-step procedure, although results in a smoothed estimate of the spiking rate, distorts the frequency content of the data. In addition, existing technique do not exploit the underlying sparsity of the frequency content of the data in favor of estimation accuracy.

In order to address these issues, we considered a model where the neuronal ensemble is driven by a harmonic second-order stationary process through a logistic link and according to Bernoulli statistics. We integrated techniques from point process modeling and spectral estimation of second-order stationary process in order to perform the PSD estimation in a Bayesian framework. Our proposed technique enjoys from several features which improve over existing techniques for obtaining spectral representations of neuronal spiking data. First, we directly estimate the PSD from spiking data by regressing the second-order statistics of the underlying process to the observed data, without any time-domain smoothing. Second, motivated by the spectral sparsity of biological signals such as EEG and LFP, we incorporated sparsity-enforcing priors in our PSD estimation. Third, we provided an algorithm for constructing confidence intervals for the PSD estimates.

We compared our proposed method with existing techniques for computing spectral representations of point processes using simulated as well as real data. As for the simulated data, we considered neuronal ensembles driven by oscillatory covariates in the form of dual-tone signals as well as autoregressive processes. Our results showed that the proposed method significantly outperforms the aforementioned existing techniques. Application of our method to multi-unit recordings from a patient undergoing anesthesia showed that the estimated PSD of the neuronal ensemble using our method exhibits a striking resemblance to the PSD of the corresponding LFP signal. This results confirms the findings of [33], which show that the spiking dynamics under general anesthesia is governed by the LFP as a salient neural covariate.

The complexity of the spectra which are identifiable from ensemble neuronal observations is limited by the average spiking rate in PSTH, i.e., the amount of available information for inference purposes. A potential limitation of spectral estimation from binary data is thus in extracting complex (and not necessarily sparse) spectral structures under low neuronal spiking rates. Nevertheless, we have demonstrated the significant performance gain of our proposed method over the existing techniques in estimating sparse narrowband spectral structures detectable under low neuronal spiking rates. This performance gain, however, comes at the cost of a higher computational complexity.

Our technique is particularly useful in analyzing the harmonic structure of spiking activity independently of the local field potentials, without any prior assumption of the spectral spread and content of the underlying neural processes. Our method can be modified in a straightforward fashion to handle other spiking models such as Poisson statistics. In addition, although we have posed the problem in the neuronal spiking data application, our algorithm can be applied to a wide variety of binary data, such as heart beat data, in order to obtain a robust spectral representation. In the spirit of easing reproducibility, we have archived a MATLAB implementation of our method on the open source repository GitHub and made it publicly available [32].

Chapter 3: Real-Time Tracking of Selective Auditory Attention from M/EEG: A Bayesian Filtering Approach

The ability to select a single speaker in an auditory scene, consisting of multiple competing speakers, and maintain attention to that speaker is one of the hallmarks of human brain function. This phenomenon has been referred to as the cocktail party effect [36–38]. The mechanisms underlying the real-time process by which the brain segregates multiple sources in a cocktail party setting has been the topic of active research for decades [39,40]. Although the details of these mechanisms are for the most part unknown, various studies have underpinned the role of specific neural processes involved in this function. As the acoustic signals propagate through the auditory pathway, they are decomposed into spectrotemporal features at different stages, and a rich representation of the complex auditory environment reaches the auditory cortex. It has been hypothesized that the perception of an auditory object is the result of adaptive binding as well as discounting of these features [41–44].

From a computational modeling perspective, there have been several attempts at designing so-called "attention decoders", where the goal is to reliably decode the attentional focus of a listener in a multi-speaker environment using non-invasive neuroimaging techniques like electroencephalography (EEG) [45–47] and magnetoencephalography (MEG) [48–52]. These methods are typically based on reverse correlation or estimating linear encoding/decoding models using off-line regression techniques, and thereby detecting salient peaks in the model coefficients that are modulated by the attentional state [53]. The aforementioned salient peaks have been observed at a typical lag of $\sim 200\,\text{ms}$ for EEG [46] and $\sim 100\,\text{ms}$ for MEG [48], implying the longer-lasting effect and further processing of the attended stimuli as compared to the unattended ones.

Although the foregoing approaches have proven successful in reliable attention decoding, they have two major limitations that make them unsuitable for emerging real-time applications such as Brain-Computer Interface (BCI) systems and smart hearing aids. First, the temporal resolution for decoding the attentional state is on the order of tens of seconds, whereas humans can switch their attention from one speaker to another at a much shorter time scale. This is due to their so-called "batch-mode" design, which requires the entire data from one or multiple trials at once for processing. Second, approaches based on linear regression (e.g., reverse correlation) need large training datasets, often from multiple subjects and trials, to estimate the decoder/encoder reliably. Access to such training data is only possible through repeated calibration stages, which may not always be possible in real-time applications. While recent results [50,51] address the first shortcoming by employing state-space models and thereby producing robust estimates of the attentional state from limited data, they are not yet suitable for real-time applications.

In this chapter, we close this gap by designing a modular framework for real-time attention decoding from non-invasive M/EEG recordings that overcomes the aforementioned limitations using techniques from Bayesian filtering. Our proposed framework includes three main modules. The first module pertains to estimating *dynamic* models of decoding/encoding in *real-time*. To this end, we use the forgetting factor mechanism of the Recursive Least Squares (RLS) algorithm together with the ℓ_1 regularization penalty from Lasso to capture the dynamics in the data while preventing overfitting [52, 54]. The real-time inference is then efficiently carried out using a Forward-Backward Splitting (FBS) procedure [55]. In the second module, we extract an attention-modulated feature, which we refer to as "attention marker", as a function of the M/EEG recordings, the estimated encoding/decoding coefficients, and the auditory stimuli. For instance, the attention marker can be a correlation-based measure or the magnitude of certain peaks in the model coefficients. We carefully design the attention marker features to capture the attention modulation and thereby maximally separate the contributions of the attended and unattended speakers in the neural response in both MEG and EEG applications.

The extracted features are then passed to a novel state-space estimator in the third module, and thereby are translated into robust and dynamic measures of the attentional state. The state-space estimator is based on Bayesian fixed-lag smoothing, and operates in *near real-time* with controllable delay. The fixed-lag design creates a trade-off between real-time operation and robustness to stochastic fluctuations. In addition, we modify the Expectation-Maximization algorithm and the nonlinear filtering and smoothing techniques of [51] for real-time implementation. Compared to existing techniques, our algorithms require minimal supervised data for initialization and tuning. In order to validate our real-time attention decoding algorithms, we apply them to both simulated and experimentally recorded EEG and MEG data in dual-speaker environments. Our results suggest that the performance of our proposed framework is comparable to the state-of-the-art batch-mode algorithms of [45,47,51], while operating in near real-time with $\sim 1\,\mathrm{s}$ delay.

The rest of the chapter is organized as follows: In Section 3.1, we develop the three main modules in our proposed framework as well as the corresponding estimation algorithms. We present the application of our framework to both synthetic and experimentally recorded M/EEG data in Section 3.2, followed by discussion and concluding remarks in Section 3.3.

3.1 Material and Methods

Figure 3.1 summarizes our proposed framework for real-time tracking of selective auditory attention from M/EEG. In the *Dynamic Encoder/Decoder Estimation* module, the encoding/decoding models are fit to neural data in real-time. The *Attention Marker* module uses the estimated model coefficients as well as the recorded data to compute a feature that is modulated by the instantaneous attentional state. Finally, in the *State-Space Model* module, the foregoing features are refined through a linear state-space model with nonlinear observations, resulting in robust and dynamic estimates of the attentional state.

In Section 3.1.1, we formally define the dynamic encoding/decoding models and develop low-complexity and real-time techniques for their estimation in Section 3.1.2. This is followed by Section 3.1.3, in which we define suitable attention markers

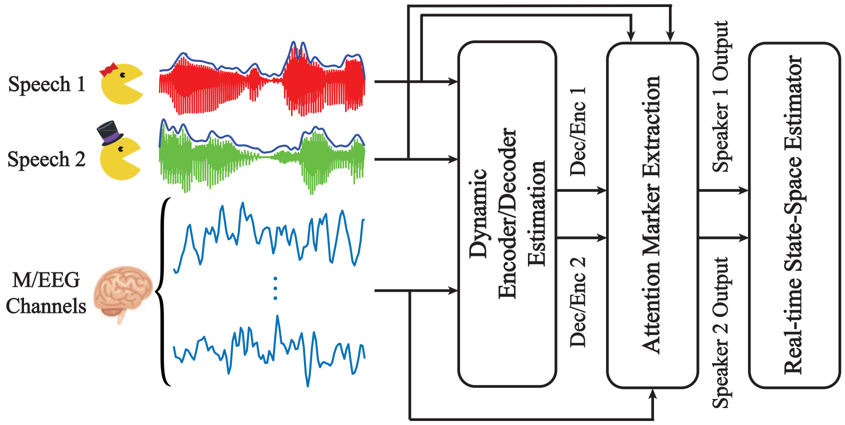
Speech 1
Speech 2
M/EEG Channels
Dynamic Encoder/Decoder Estimation
Dec/Enc 1
Dec/Enc 2
Attention Marker Extraction
Speaker 1 Output
Speaker 2 Output
Real-time State-Space Estimator

Figure 3.1: A schematic depiction of our proposed framework for real-time tracking of selective auditory attention from M/EEG.

for M/EEG inspired by existing literature. In Section 3.1.4, we propose a state-space model that processes the extracted attention markers in order to produce near real-time estimates of the attentional state with minimal delay, and we discuss its estimation procedure in Section 3.1.5.

3.1.1 Dynamic Encoding and Decoding Models: definition

The role of a neural encoding model is to map the stimulus to the neural response. Inspired by existing literature on attention decoding [45, 48, 51], we take the speech envelopes as covariates representing the stimuli. The neural response is manifested in the M/EEG recordings. Encoding models can be used to predict the neural response from the stimulus. In contrast, in a neural decoding model, the goal is to express the stimulus as a function of the neural response. Inspired by previous studies, we consider linear encoding and decoding models in this work.

The encoding and decoding models can be cast as mathematically dual formulations. In a dual-speaker environment, let $s_t^{(1)}$ and $s_t^{(2)}$ denote the speech envelopes (in logarithmic scale), corresponding to speakers 1 and 2, respectively, for $t = 1, 2, \ldots, T$. Also, let e_t^c denote the neural response recorded at time t and channel c, for $c = 1, 2, \ldots, C$. Throughout the chapter, we assume the same sampling frequency for both the M/EEG channels and the envelopes. Consider consecutive and non-overlapping windows of length W, and define $K := \lfloor \frac{T}{W} \rfloor$. We consider piece-wise constant dynamics for the encoding and decoding coefficients, in which the coefficients assume to be constant over each window.

In the encoding setting, we define the vector $\mathbf{s}_t^{(i)} := [s_t^{(i)}, s_{t-1}^{(i)}, \ldots, s_{t-L_e}^{(i)}]^\top$ for $i = 1, 2$, where L_e is the total lag considered in the model. Also, let E_t denote a generic linear combination of $e_t^1, e_t^2, \ldots, e_t^C$ with some fixed set of weights. These weights can be set to select a single channel, i.e., $E_t = e_t^c$ for some c, or they can be pre-estimated from training data so that E_t represents the dominant auditory component of the neural response [56]. The encoding coefficients then relate $\mathbf{s}_t^{(i)}$ to E_t. In the decoding setting, we define the vector $\mathbf{e}_t := [e_t^1, e_t^2, \ldots, e_t^C]^\top$ and $\boldsymbol{\mathcal{E}}_t := \left[1, \mathbf{e}_t^\top, \mathbf{e}_{t+1}^\top, \ldots, \mathbf{e}_{t+L_d}^\top\right]^\top$, where L_d is the total lag in the decoding model and determines the extent of future neural responses affected by the current stimuli. The decoding coefficients then relate $\boldsymbol{\mathcal{E}}_t$ to $s_t^{(i)}$.

Our goal is to recursively estimate the encoding/decoding coefficients in a real-time fashion as the new data samples become available. In addtion, we aim to simultaneously induce adaptivity of the parameter estimates and capture their sparsity. To this end, we employ the following generic optimization problem:

$$\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}} \sum_{j=1}^{k} \lambda^{k-j} \| \mathbf{y}_j - \mathbf{X}_j \boldsymbol{\theta} \|_2^2 + \gamma \| \boldsymbol{\theta} \|_1 , \quad k = 1, 2, \ldots, K \tag{3.1}$$

where $\mathbf{y}_j$ and $\mathbf{X}_j$ are the vector of response variables and the matrix of covariates pertinent to window j, $\boldsymbol{\theta}$ is the parameter vector, $\lambda \in (0, 1]$ is the forgetting factor, and γ is a regularization parameter. The optimization problem of Eq. 3.1 is a modified version of the LASSO problem [57].

For the encoding problem, we define $\mathbf{y}_k := \left[E_{(k-1)W+1}, E_{(k-1)W+2}, \ldots, E_{kW}\right]^\top$ and $\mathbf{X}_k^{(i)} := \left[\mathbf{s}_{(k-1)W+1}^{(i)}, \mathbf{s}_{(k-1)W+2}^{(i)}, \ldots, \mathbf{s}_{kW}^{(i)}\right]^\top$, for $k = 1, 2, \ldots, K$ and $i = 1, 2$. Therefore, the full encoding covariate matrix at the k^{th} window is defined as $\mathbf{X}_k := \left[\mathbb{1}_{W\times 1}, \mathbf{X}_k^{(1)}, \mathbf{X}_k^{(2)}\right]$, where the all-ones vector $\mathbb{1}_{W\times 1}$ corresponds to the regression intercept. In the decoding problem, we define $\mathbf{y}_k := \left[s_{(k-1)W+1}^{(i)}, s_{(k-1)W+2}^{(i)}, \ldots, s_{kW}^{(i)}\right]^\top$, where $i \in \{1, 2\}$. Also, the full decoding covariate matrix at the k^{th} window is $\mathbf{X}_k := \left[\boldsymbol{\mathcal{E}}_{(k-1)W+1}, \boldsymbol{\mathcal{E}}_{(k-1)W+1}, \ldots, \boldsymbol{\mathcal{E}}_{kW}\right]^\top$, for $k = 1, 2, \ldots, K$.

The optimization problem of Eq. (3.1) has a useful Bayesian interpretation: if the observation noise were i.i.d. Gaussian, and the parameters were exponentially distributed, it is akin to the maximum *a posteriori* (MAP) estimate of the parameters. The quadratic terms correspond to the exponentially-weighted log-likelihood of the observations up to window k, and the ℓ_1-norm corresponds to the log-density of an independent exponential prior on the elements of $\boldsymbol{\theta}$. The exponential prior serves as an effective regularization to promote sparsity of the estimate $\hat{\boldsymbol{\theta}}_k$. Note that we have $\boldsymbol{\theta} \in \mathbb{R}^{1+2(L_e+1)}$ for the encoding model and $\boldsymbol{\theta} \in \mathbb{R}^{1+C(L_d+1)}$ for the decoding model in (3.1).

Remark 3.1. The hyperparameter λ provides a tradeoff between the adaptivity and the robustness of estimated coefficients, and it can be determined based on the inherent dynamics in the data. The case of $\lambda = 1$ corresponds to the natural data log-likelihood, i.e., the batch-mode parameter estimates. It has been shown that $\frac{W}{1-\lambda}$ can serve as the *effective* number of recent samples used to calculate $\hat{\boldsymbol{\theta}}_k$ in (3.1) [58]. The parameter $\frac{W}{1-\lambda}$ can also be viewed as the dynamic integration time: it needs to be chosen long enough so that the estimation is stable, but also short enough to be able to capture the dynamics of neural process involved in switching attention. The hyperparameter γ controls the tradeoff between the Maximum Likelihood (ML) fit and the sparsity of estimated coefficients, and it is usually determined through cross-validation.

Remark 3.2. In the decoding problem, Eq. (3.1) is solved separately at each window for each speech envelope, resulting in a set of decoding coefficients per speaker. In the encoding setting, we combine the stimuli as explained and solve Eq. (3.1) once at each window to obtain both of the encoder estimates. If the encoding/decoding coefficients are expected to be sparse in a basis represented by the columns of a matrix $\mathbf{G}$, such as the Haar or Gabor bases, we can replace $\mathbf{X}_j$ in (3.1) by $\mathbf{X}_j\mathbf{G}$, for $j = 1, 2, \ldots, k$, and solve for $\hat{\boldsymbol{\theta}}_k$ as before. Then, the final encoding/decoding coefficients are given by $\mathbf{G}\hat{\boldsymbol{\theta}}_k$. In the context of encoding models, the coefficients are referred to as the Temporal Response Function (TRF) [48, 52]. The TRFs are known to exhibit some degree of sparsity and smoothness in the lag domain, which can be represented over a basis consisting of shifted Gaussian kernels (see [52] for details).

Remark 3.3. Throughout the chapter, we assume that the envelopes of the clean speeches are available. Given that this assumption does not hold in practical scenarios, recent algorithms on the extraction of speech envelopes from acoustic mixtures [59–63] can be added as a pre-processing module to our framework.

3.1.2 Dynamic Encoding and Decoding Models: parameter estimation

There are several standard optimization techniques that can be used to find the minimizer in (3.1). Off-line algorithms such as interior point methods do not meet the real-time requirements of our dynamic estimation. The SPARLS algorithm has been introduced in [64] to solve the problem in (3.1) through EM iterations, and it has been successfully adopted in [52] to estimate encoding coefficients in a dynamic fashion. However, the EM algorithm and the constant step-size in SPARLS may result in low convergence rates. Hence, to adapt our estimation procedure for real-time applications, we use the Forward-Backward Splitting (FBS) method [55], also known as the proximal gradient method, to solve for $\hat{\boldsymbol{\theta}}_k$ in (3.1). FBS is suited for optimization problems where the objective function can be expressed as the sum of a differentiable term, e.g., the log-likelihood term in (3.1), and a simple non-differentiable term, e.g., the ℓ_1-norm in (3.1). This type of problems frequently arise in signal processing and machine learning [65–67].

In summary, each FBS iteration for the optimization problem in (3.1) includes two steps: 1) taking a descent step along the gradient of the log-likelihood term, and 2) applying a soft-thresholding shrinkage operator [58,68]. This procedure provides an algorithm that uses recursive and low-complexity updates in an online fashion to solve Eq. (3.1) upon the arrival of a new data window. The optimization problem in (3.1) can be rewritten as:

$$\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}} \ \boldsymbol{\theta}^T \mathbf{A}_k \boldsymbol{\theta} + \mathbf{b}_k^T \boldsymbol{\theta} + \gamma \parallel \boldsymbol{\theta} \parallel_1, \quad k = 1, 2, \ldots, K, \tag{3.2}$$

where $\mathbf{A}_k$ and $\mathbf{b}_k$ can be updated recursively. Algorithm 5 summarizes the steps of the FBS algorithm to solve for $\boldsymbol{\theta}_k$ in (3.1), when moving from window $k-1$ to window k, as well as the required recursive update rules for $\mathbf{A}_k$ and $\mathbf{b}_k$. The parameter $\mathcal{S}_{FBS}$ in Algorithm 5 denotes the stopping condition for the FBS algorithm, which can be a maximum iteration number or a convergence criterion on the objective function.

Algorithm 5 Parameter Estimation in Dynamic Encoding and Decoding Models by Forward-Backward Splitting

```
Inputs: $\mathbf{y}_k$, $\mathbf{X}_k$, $\hat{\boldsymbol{\theta}}_{k-1}$, $\mathbf{A}_{k-1}$, $\mathbf{b}_{k-1}$, $\lambda$, $\gamma$, $\mathcal{S}_{FBS}$.
Output: $\hat{\boldsymbol{\theta}}_k$, $\mathbf{A}_k$, $\mathbf{b}_k$.
$\mathbf{A}_k = \lambda \mathbf{A}_{k-1} + \mathbf{X}_k^T \mathbf{X}_k$
$\mathbf{b}_k = \lambda \mathbf{b}_{k-1} - 2\mathbf{X}_k^T \mathbf{y}_k$
initialize $\boldsymbol{\theta}$ with $\hat{\boldsymbol{\theta}}_{k-1}$
while $\neg \mathcal{S}_{FBS}$ do
    choose stepsize $\tau$
    $\mathbf{u} = \boldsymbol{\theta} - \tau \left(2\mathbf{A}_k \boldsymbol{\theta} + \mathbf{b}_k\right)$
    $\boldsymbol{\theta}_i = \text{sign}(\mathbf{u}_i) \times \max\left\{|\mathbf{u}_i| - \gamma\tau, 0\right\}$, for each element of $\boldsymbol{\theta}$
end while
$\hat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}$
```

Remark 3.4. A proper step-size choice in Alg. 5 at each FBS iteration is crucial to the convergence of the algorithm. For a fixed step-size, it has been shown that

$\tau < \frac{2}{L(\nabla f_k)}$ ensures the stability and convergence of the algorithm [55], where $L(.)$ represents the Lipschitz constant, and f_k represents the log-likelihood term in (3.1). Through standard Cauchy-Schwarz and triangle inequality manipulations, we can calculate the simple upper bound $L(\nabla f_k) \leq L_{\text{ub}} = 2\sum_{j=1}^{k} \lambda^{k-j} \operatorname{trace}\left\{\mathbf{X}_k^T \mathbf{X}_k\right\}$, implying that $\tau < \frac{2}{L_{\text{ub}}}$ ensures stability; however, this loose upper bound may decrease the convergence rate of the algorithm. Thus, it is more beneficial to ensure stability through backtracking and employing acceleration schemes such as adaptive step-size selection or the Nesterov's method [69]. In this work, we have used the FASTA software package [69] available online [70], which has built-in features for all the foregoing FBS step-size adjustment methods.

3.1.3 Attention Markers

We define the *attention marker* as a mapping function from the estimated encoding/decoding coefficients for each speaker as well as the data in each window to positive real numbers. To be more precise, at window k and for speaker i, in the context of encoding models, the attention marker takes the speaker's estimated encoding coefficients $\hat{\boldsymbol{\theta}}_k^{(i)}$, the speaker's covariate matrix $\mathbf{X}_k^{(i)}$, and the M/EEG responses $\mathbf{y}_k$ as inputs; similarly, in the context of decoding models, the attention marker takes the speaker's estimated decoding coefficients $\hat{\boldsymbol{\theta}}_k^{(i)}$, the M/EEG covariate matrix $\mathbf{X}_k$, and the speaker's speech envelope vector $\mathbf{y}_k^{(i)}$ as inputs. In both cases, the attention marker outputs a positive real number, which we denote by $m_k^{(i)}$ henceforth, for $i = 1, 2$ and $k = 1, 2, \ldots, K$. Thus, in the modular design of Fig.

3.1, at each window k, the two outputs $m_k^{(1)}$ and $m_k^{(2)}$ are passed from the Attention Maker module to the State-Space Model module as measures of the attentional state at window k.

In [45], a correlation-based measure has been adopted in the decoding model to classify the attended and the unattended speeches in a dual-speaker environment. The approach in [45] is based on estimating an *attended* decoder from the training data to reconstruct the speech envelope from EEG for each trial. Then, the correlation of this reconstructed envelope with each of the two speech envelopes is computed, and the speaker with the larger correlation coefficient is deemed as the attended speaker. This method cannot be directly applied to the real-time setting, since the lack of abundant training data hinders a reliable estimate of the *attended* decoder. However, assuming that the auditory M/EEG response is more influenced by the attended speaker than the unattended one, we can expect that the decoder corresponding to the *attended* speaker exhibits a higher performance in reconstructing the speech envelope it has been trained on, as suggested by the classification comparisons in [45]. Inspired by these results, we can define the attention marker in the decoding scenario as the correlation magnitude between the speech envelope and its reconstruction by the corresponding decoder, i.e., $m_k^{(i)} = f\left(\hat{\boldsymbol{\theta}}_k^{(i)}, \mathbf{X}_k, \mathbf{y}_k^{(i)}\right) := \left|\text{corr}\left(\mathbf{y}_k^{(i)}, \mathbf{X}_k \hat{\boldsymbol{\theta}}_k^{(i)}\right)\right|$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$. As we will demonstrate later in Section 3.2, this attention marker is suitable for the analysis of EEG recordings.

In the context of cocktail party studies using MEG, it has been shown that the magnitude of the negative peak in the TRF of the attended speaker around a lag of

100 ms, referred to as the M100 component, is higher than that of the unattended speaker [48, 51, 52]. Inspired by these findings, in the encoding scenario applied to MEG data, we can define the attention marker $m_k^{(i)}$ to be the magnitude of the $\hat{\boldsymbol{\theta}}_k^{(i)}$ coefficients corresponding to the M100 component, for $i = 1, 2$ and $k = 1, 2, \ldots, K$.

Due to the inherent uncertainties in the M/EEG recordings, the limitations of non-invasive neuroimaging in isolating the relevant neural processes, and the unknown and likely nonlinear processes involved in auditory attention, the foregoing attention markers derived from linear models are not readily reliable indicators of the attentional state. Given ample training data, however, these attention markers have been validated using batch-mode analysis. However, their usage in a real-time setting requires more care, as the limited data in real-time applications adds a major source of uncertainty to the foregoing list. To address this issue, a state-space model is required in the real-time setting to correct for the uncertainties and stochastic fluctuations of the attention markers caused by the limited integration time in real-time application. We will discuss in detail the formulation and advantages of such a state-space model in the following subsection.

3.1.4 Dynamic State-Space Model: definition

In order to translate the attention markers $m_k^{(1)}$ and $m_k^{(2)}$, for $k = 1, 2, \ldots, K$, into a robust and statistically interpretable measure of the attentional state, we employ state-space models. Inspired by the models used in [51], we design a new state-space model and a corresponding estimator that operates in a fixed-lag smoothing fashion, and thereby admits real-time processing while maintaining the benefits of batch-mode state-space models. Recall that the index k corresponds to a window in time ranging from $t = (k-1)W + 1$ to $t = kW$; however, we refer to each index k as an *instance* when talking about the state-space model not to be confused with the sliding window of the fixed-lag design.

Figure 3.2 displays the fixed-lag smoothing design of the state-space estimator. Suppose that we are at the instance $k = k_0$. We consider a window of length $K_W = K_B + K_F + 1$ as shown in Fig. 3.2, where K_F and K_B are respectively called the forward-lag and the backward-lag. In order to carry out the computations in real-time, we assume all of the attentional state estimates to be fixed prior to this window and only update our estimates for the instances within, based on $m_k^{(1)}$'s and $m_k^{(2)}$'s inside the window. In a fixed-lag framework, at $k = k_0$, the goal is to provide an estimate of the attentional state at instance $k = k^*$, where $k^* = k_0 - K_F$. The parameter K_F creates a tradeoff between real-time and robust estimation of the attentional state. For $K_F = 0$, the estimation is carried out fully in real-time; however, the estimates lack robustness to the fluctuations of the outputs of the attention marker block. The backward-lag K_B incorporates the information before k^* in order to make the estimates more reliable, and controls the computational cost of the state-space model for fixed values of K_F. Throughout the rest of the chapter, we use the expression *real-time* for referring to algorithms that operate with a fixed forward-lag of K_F. We will discuss specific choices of K_F and K_B and their implications in Section 3.2.

Suppose we are in a window of length K_W where the instances are indexed

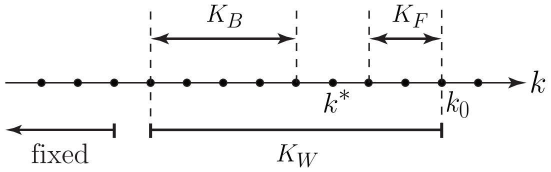

Figure 3.2: The parameters involved in state-space fixed-lag smoothing.

by $k = 1, 2, \ldots, K_W$. Inspired by [51], we assume a linear state-space model on the logit-probability of attending to speaker 1. We define the binary random variable $n_k = 1$ when speaker 1 is attended and $n_k = 2$ when speaker 2 is attended, at instance k. The goal is to obtain estimates of $p_k := \mathrm{P}\,(n_k = 1)$ together with its confidence intervals for $1 \leq k \leq K_W$. The state dynamics are given by:

$$\begin{cases} p_k = \mathrm{P}\,(n_k = 1) = 1 - \mathrm{P}\,(n_k = 2) = \frac{1}{1+\exp(-z_k)} \\ z_k = c_0 z_{k-1} + w_k \\ w_k \sim \mathcal{N}(0, \eta_k) \\ \eta_k \sim \text{Inverse-Gamma}\,(a_0, b_0) \end{cases} \tag{3.3}$$

The dynamics of the main latent variable z_k are controlled by its transition scale c_0 and state variance η_k. The hyperparameter $0 \leq c_0 \leq 1$ ensures the stability of the updates for z_k. The state variance η_k is modeled using an Inverse-Gamma conjugate prior with hyper-parameters a_0 and b_0. The log-prior of the Inverse-Gamma density takes the form $\ln \mathrm{P}\,(\eta_k) = -(a_0 + 1) \ln \eta_k - \frac{b_0}{\eta_k} + C$ for $\eta_k > 0$, where C is a normalization constant. By choosing a_0 greater and sufficiently close to 2, the variance of the Inverse-Gamma distribution takes large values and therefore can serve as a non-informative conjugate prior. Considering the fact that we do not expect the attentional state to have high fluctuations within a small window of time, we can further tune the hyperparameters a_0 and b_0 for the prior to promote smaller values of η_k's. This way, we can avoid large consecutive fluctuations of the z_k's, and consequently the p_k's.

Next, we develop an observation model relating the state dynamics of Eq. (3.3) to the observations $m_k^{(1)}$ and $m_k^{(2)}$ for $k = 1, 2, \ldots, K_W$. To this end, we use the latent variable n_k as the link between the states and observations:

$$\begin{cases} \begin{cases} m_k^{(i)} \,\Big|\, n_k = i \sim \text{Log-Normal}\left(\rho^{(a)}, \mu^{(a)}\right) \\[2ex] m_k^{(i)} \,\Big|\, n_k \neq i \sim \text{Log-Normal}\left(\rho^{(u)}, \mu^{(u)}\right) \end{cases} , \quad i = 1, 2 \\[2ex] \rho^{(a)} \sim \text{Gamma}\left(\alpha_0^{(a)}, \beta_0^{(a)}\right), \quad \mu^{(a)} \,\Big|\, \rho^{(a)} \sim \mathcal{N}\left(\mu_0^{(a)}, \rho^{(a)}\right) \\[2ex] \rho^{(u)} \sim \text{Gamma}\left(\alpha_0^{(u)}, \beta_0^{(u)}\right), \quad \mu^{(u)} \,\Big|\, \rho^{(u)} \sim \mathcal{N}\left(\mu_0^{(u)}, \rho^{(u)}\right) \end{cases} \tag{3.4}$$

When speaker $i = 1, 2$ is attended to, we use a Log-Normal distribution on $m_k^{(i)}$'s, with log-prior given by $\ln \mathrm{P}\left(m_k^{(i)} \,\middle|\, n_k = i\right) = -\ln m_k^{(i)} + \frac{1}{2}\ln \rho^{(a)} - \frac{\rho^{(a)}}{2}\left(\ln m_k^{(i)} - \mu^{(a)}\right)^2 + C^{(i)}$, where $\mu^{(a)} \in \mathbb{R}$, $\rho^{(a)} \in \mathbb{R}_{>0}$, and $C^{(i)}$ is a normalization constant, for $i = 1, 2$, and $k = 1, 2, \ldots, K_W$. Similarly, when speaker $i = 1, 2$ is *not* attended to, we use a Log-Normal prior on $m_k^{(i)}$ with parameters $\rho^{(u)}$ and $\mu^{(u)}$. As mentioned before, choosing an appropriate attention marker results in a statistical separation between $m_k^{(1)}$ and $m_k^{(2)}$, if only one speaker is attended. The Log-Normal distribution is a distribution on $\mathbb{R}_{>0}$ which lets us capture this concentration in the values of $m_k^{(i)}$'s.

In contrast to [51], this distribution also leads to closed form update rules, which significantly reduces computational costs. We have also imposed conjugate priors on the joint distribution of (ρ, μ)'s, which factorizes as $\ln \mathrm{P}(\rho, \mu) = \ln \mathrm{P}(\rho) + \ln \mathrm{P}(\mu \,|\, \rho)$. The hyperparameters α_0, β_0, and μ_0 serve to tune the attended and the unattended Log-Normal distributions to create separation between the attended and unattended cases. These hyperparameters can be determined based on the mean and variance information of $m_k^{(i)}$'s in a supervised manner, where the attended speaker is known.

The parameters of the state-space model are therefore $\mathbf{\Omega} = \big\{z_{1:K_W}, \eta_{1:K_W}, \rho^{(a)}, \mu^{(a)}, \rho^{(u)}, \mu^{(u)}\big\}$, which have to be inferred from $m^{(1)}_{1:K_W}$ and $m^{(2)}_{1:K_W}$. As mentioned before, our goal in the fixed-lag smoothing approach is to estimate z_{k^*} and η_{k^*} in each window, where $k^* = K_W - K_F$. However, in order to do so in our model, we perform the inference step over all the parameters in $\mathbf{\Omega}$ and output the estimates of $\{z_{k^*}, \eta_{k^*}\} \in \mathbf{\Omega}$. The estimated $\mathbf{\Omega}$ would then serve as the initialization for parameter estimation in the next window.

Remark 3.5. The state-space models given in Eqs. 3.3 and 3.4 have two major differences with the one used in [51]. First, in [51], the distribution over the correlative measure for the *unattended* speaker is assumed to be uniform. However, this assumption may not hold for other attention markers in general. For instance, the M100 magnitude of the TRF estimated from MEG data is a positive random variable, which is concentrated on higher values for the attended speaker compared to the unattended speaker. In order to address this issue, we consider a parametric distribution in Eq. (3.4) over the attention marker corresponding to the unattended speaker and infer its parameters from the data. If this distribution is indeed uniform and non-informative, the variance of the unattended distribution, which is estimated from the data, would be large enough to capture the flatness of the distribution. Second, the parametrization of the observations using Log-Normal densities and their corresponding priors factorized using Gamma and Gaussian priors, admits fast and closed-form update equations in the real-time setting. As we will show in Section 3.1.5, these models also have the advantage of incorporating low-complexity updates by simplifying the EM procedure. In addition, the Log-Normal distribution as a generic unimodal distribution allows us to model a larger class of attention markers.

3.1.5 Dynamic State-Space Model: parameter estimation

For notational simplicity, hereafter we use the boldface version of a variable to denote a vector containing all its instances, e.g., $\boldsymbol{z} := z_{1:K_W}$ and $\boldsymbol{m}^{(i)} := m^{(i)}_{1:K_W}$ for $i = 1, 2$. The inference problem for $\boldsymbol{\Omega}$ from $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ can be expressed as:

$$\widehat{\boldsymbol{\Omega}} = \arg\max_{\boldsymbol{\Omega}} \ \ln \mathrm{P}\left(\boldsymbol{\Omega} \,\middle|\, \boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}\right) = \arg\max_{\boldsymbol{\Omega}} \ \ln \mathrm{P}\left(\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)} \,\middle|\, \boldsymbol{\Omega}\right) + \ln \mathrm{P}\left(\boldsymbol{\Omega}\right), \tag{3.5}$$

where the log-likelihood and the log-prior are respectively expanded as:

$$\ln \mathrm{P}\left(\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)} \,\middle|\, \boldsymbol{\Omega}\right) = \ln \left(\sum_{n_{1:K_W}} \sum_{k=1}^{K_W} p_k \, \mathrm{P}\left(m_k^{(1)} \,\middle|\, n_k, \boldsymbol{\Omega}\right) \mathrm{P}\left(m_k^{(2)} \,\middle|\, n_k, \boldsymbol{\Omega}\right) \right), \tag{3.6}$$

$$\ln \mathrm{P}(\boldsymbol{\Omega}) = \ln \mathrm{P}\left(\rho^{(a)}, \mu^{(a)}\right) + \ln \mathrm{P}\left(\rho^{(u)}, \mu^{(u)}\right) + \underbrace{\sum_{k=1}^{K_W} \left[-\frac{1}{2}\ln \eta_k - \frac{(z_k - c_0 z_{k-1})^2}{2\eta_k} + \ln \mathrm{P}(\eta_k)\right]}_{\ln \mathrm{P}(\boldsymbol{z},\boldsymbol{\eta})} + \text{cnst}. \tag{3.7}$$

Similar to the treatment in [51], we use an Expectation Maximization (EM) algorithm with $\boldsymbol{n}$ as the latent variables to infer $\boldsymbol{\Omega}$. Note that the optimization problem in (3.5) is non-convex in general; thus, the choice of initial conditions and hyperparameters for priors are important for reaching a desirable local maximum. Having the estimate $\widehat{\boldsymbol{\Omega}}^{(\ell)}$ for $\boldsymbol{\Omega}$ at the ℓ^{th} EM iteration, we will next derive the E-step and M-step of the $(\ell+1)^{\text{th}}$ EM iteration.

3.1.5.1 The E-step

In the E-step, the surrogate function $Q\left(\boldsymbol{\Omega} \,\middle|\, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ is calculated as:

$$Q\left(\boldsymbol{\Omega} \,\middle|\, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) = \frac{1}{K_W} \underbrace{\mathbb{E}\left\{\ln \mathrm{P}\left(\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \boldsymbol{n} \,\middle|\, \boldsymbol{\Omega}\right)\right\}}_{\mathcal{A}} + \ln \mathrm{P}(\boldsymbol{\Omega}), \tag{3.8}$$

where the expectation of the *complete* log-likelihood $\ln \mathrm{P}\left(\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \boldsymbol{n} \,\middle|\, \boldsymbol{\Omega}\right)$ needs to be calculated with respect to $\boldsymbol{n}$ given $\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}$. For notational simplicity, hereafter we drop the $\boldsymbol{n} \,\big|\, \boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}$ subscript of the conditional expectations.

We have used a *normalized* version of the log-likelihood in Eq. (3.8) for two reasons. First, the window length K_W is a hyperparameter in our framework, which we can modify to find the optimal trade-off between the dimensionality of the state-space and history-dependence of the model. Thus, to change the window length for fixed priors, it is important to normalize the contribution of the log-likelihood in (3.8). Second, as noted before, we have a non-convex inference problem, which makes the resulting local maximum dependent on the conjugate priors used. We can use samples of $m_k^{(i)}$'s to estimate the attended and the unattended Log-Normal distributions and tune the hyperparameters to these distributions. By normalizing the log-likelihood term, we are enforcing informative and empirical prior distributions which would guide the inference procedure towards a plausible local maximum. For instance, for the correlation-based attention marker, we expect that a plausible solution would result in the attended Log-Normal distribution being concentrated around larger correlation values compared to the unattended distribution. Nevertheless, the forthcoming derivations can be carried out without the normalization factor $1/K_W$ in a similar fashion.

Let $\mathbb{I}_u(v)$ represent the indicator function, i.e., it is equal to one if $v = u$ and zero otherwise. Conditioning on $\boldsymbol{n}$ and using the conditional independence of $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ given $\boldsymbol{n}$ and $\boldsymbol{\Omega}$, the expected log-likelihood $\mathcal{A}$ in (3.8) can be simplified as:

$$
\begin{aligned}
\mathcal{A} &= \sum_{i=1}^{2} \mathbb{E}\left\{\ln \mathrm{P}\left(\boldsymbol{m}^{(i)} \,\middle|\, \boldsymbol{n}, \boldsymbol{\Omega}\right)\right\} + \mathbb{E}\left\{\ln \mathrm{P}\left(\boldsymbol{n} \,\middle|\, \boldsymbol{\Omega}\right)\right\} \\
&= \sum_{k=1}^{K_W} \left[\sum_{i=1}^{2} \mathbb{E}\left\{\ln \mathrm{P}\left(m_k^{(i)} \,\middle|\, n_k, \boldsymbol{\Omega}\right)\right\} + \mathbb{E}\left\{\ln \mathrm{P}\left(n_k \,\middle|\, \boldsymbol{\Omega}\right)\right\} \right] \\
&= \sum_{k=1}^{K_W} \Bigg[\sum_{i=1}^{2} \sum_{j=1}^{2} \mathbb{E}\left\{\mathbb{I}_j(n_k)\right\} \ln \mathrm{P}\left(m_k^{(i)} \,\middle|\, n_k = j, \boldsymbol{\Omega}\right) \\
&\qquad + \underbrace{\mathbb{E}\left\{\mathbb{I}_1(n_k)\right\} p_k + \mathbb{E}\left\{\mathbb{I}_2(n_k)\right\} (1 - p_k)}_{\mathbb{E}\left\{\ln \mathrm{P}\left(n_k \,\middle|\, \boldsymbol{\Omega}\right)\right\}} \Bigg].
\end{aligned}
\tag{3.9}
$$

Note that $m_k^{(i)} \mid n_k, \boldsymbol{\Omega}$ pertains to either the attended or unattended Log-Normal distributions in Eq. (3.4) depending on the values of i and n_k. Considering that the n_k's are binary random variables and the expectations are with respect to $\boldsymbol{n} \mid \boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}$, the term $\mathbb{E}\left\{\mathbb{I}_j(n_k)\right\}$ can be computed for $j = 1, 2$ using Bayes' rule and conditional independence as:

$$\begin{aligned}
\mathbb{E}\left\{\mathbb{I}_j(n_k)\right\} &= \mathrm{P}\left(n_k = j \mid \boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \qquad (3.10)\\
&= \mathrm{P}\left(n_k = j \mid m_k^{(1)}, m_k^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)\\
&= \frac{\mathrm{P}\left(m_k^{(1)}, m_k^{(2)} \mid n_k = j, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \mathrm{P}\left(n_k = j \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)}{\mathrm{P}\left(m_k^{(1)}, m_k^{(2)} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)}\\
&= \frac{\mathrm{P}\left(m_k^{(1)} \mid n_k = j, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \mathrm{P}\left(m_k^{(2)} \mid n_k = j, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \mathrm{P}\left(n_k = j \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)}{\sum_{n_k} \mathrm{P}\left(m_k^{(1)} \mid n_k, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \mathrm{P}\left(m_k^{(2)} \mid n_k, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) \mathrm{P}\left(n_k \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)}.
\end{aligned}$$

The parameters of the Log-Normal distributions for $m_k^{(i)} \mid n_k, \widehat{\boldsymbol{\Omega}}^{(\ell)}$ are determined from the estimated $\left(\rho^{(a)}, \mu^{(a)}, \rho^{(u)}, \mu^{(u)}\right)$ in the previous EM iteration, i.e., $\widehat{\boldsymbol{\Omega}}^{(\ell)}$. Also, $\mathrm{P}\left(n_k \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) = \frac{1}{1+\exp\left(-\hat{z}_k^{(\ell)}\right)}$ in (3.10), where $\hat{z}_k^{(\ell)}$ is the estimate of z_k from the previous EM iteration. Note that $\mathbb{E}\left\{\mathbb{I}_1(n_k)\right\} = 1 - \mathbb{E}\left\{\mathbb{I}_2(n_k)\right\}$ as n_k is a binary random variable. Defining $\epsilon_k^{(\ell)} \coloneqq \mathbb{E}\left\{\mathbb{I}_1(n_k)\right\}$ with the expectation over $n_k \mid m_k^{(1)}, m_k^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}$, we can conclude the E-step by simplifying $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ in Eq. (3.8) as:

$$
\begin{aligned}
Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right) = \sum_{k=1}^{K_W} \Bigg\{ & - \rho^{(a)} \left[\epsilon_k^{(\ell)} \left(\ln m_k^{(1)} - \mu^{(a)}\right)^2 + \left(1 - \epsilon_k^{(\ell)}\right) \left(\ln m_k^{(2)} - \mu^{(a)}\right)^2\right] \\
& - \rho^{(u)} \left[\left(1 - \epsilon_k^{(\ell)}\right) \left(\ln m_k^{(1)} - \mu^{(u)}\right)^2 + \epsilon_k^{(\ell)} \left(\ln m_k^{(2)} - \mu^{(u)}\right)^2\right] \\
& + \ln \rho^{(a)} + \ln \rho^{(u)} \Bigg\} \frac{1}{2K_W} \\
- \rho^{(a)} & \left[\beta_0^{(a)} + 0.5 \left(\mu^{(a)} - \mu_0^{(a)}\right)^2\right] + \left(\alpha_0^{(a)} - 0.5\right) \ln \rho^{(a)} \\
- \rho^{(u)} & \left[\beta_0^{(u)} + 0.5 \left(\mu^{(u)} - \mu_0^{(u)}\right)^2\right] + \left(\alpha_0^{(u)} - 0.5\right) \ln \rho^{(u)} \\
+ \sum_{k=1}^{K_W} & \Bigg\{ \epsilon_k^{(\ell)} p_k + \left(1 - \epsilon_k^{(\ell)}\right) (1 - p_k) - (a_0 + 1.5) \ln \eta_k \\
& - \frac{1}{\eta_k} \left[b_0 + 0.5(z_k - c_0 z_{k-1})^2\right] \Bigg\} + \mathsf{cnst.}
\end{aligned}
\tag{3.11}
$$

where the $\mathsf{cnst.}$ term includes all the terms that are independent of $\boldsymbol{\Omega}$.

3.1.5.2 The M Step

In the M step, we maximize $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ in Eq. (3.11) with respect to $\boldsymbol{\Omega}$. The maximizers form the parameter updates for the $(\ell{+}1)^{\text{th}}$ EM iteration. As we observe in Eq. (3.11), having $\boldsymbol{n}$ as the latent variables separates the terms in $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ depending on the distribution parameters, i.e., $\left(\rho^{(a)}, \mu^{(a)}, \rho^{(u)}, \mu^{(u)}\right)$, and the terms depending on the state-space parameters, i.e., $\boldsymbol{z}$ and $\boldsymbol{\eta}$. The derivation of the update rules for the distribution parameters is straightforward through taking the derivatives of $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ and solving for their joint zero-crossings. Consequently, the closed-form formulas for the distribution parameters maximizing $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ can be expressed as:

$$\mu^{(a)^*} = \frac{1}{2}\left\{\mu_0^{(a)} + \frac{1}{K_W}\sum_{k=1}^{K_W}\left[\epsilon_k^{(\ell)}\ln m_k^{(1)} + \left(1-\epsilon_k^{(\ell)}\right)\ln m_k^{(2)}\right]\right\}, \tag{3.12}$$

$$\mu^{(u)^*} = \frac{1}{2}\left\{\mu_0^{(u)} + \frac{1}{K_W}\sum_{k=1}^{K_W}\left[\left(1-\epsilon_k^{(\ell)}\right)\ln m_k^{(1)} + \epsilon_k^{(\ell)}\ln m_k^{(2)}\right]\right\}, \tag{3.13}$$

$$\rho^{(a)^*} = \frac{2K_W\alpha_0^{(a)}}{\sum\limits_{k=1}^{K_W}\left[\epsilon_k^{(\ell)}\left(\ln m_k^{(1)}-\mu^{(a)^*}\right)^2+\left(1-\epsilon_k^{(\ell)}\right)\left(\ln m_k^{(2)}-\mu^{(a)^*}\right)^2\right]+K_W\left[2\beta_0^{(a)}+\left(\mu^{(a)^*}-\mu_0^{(a)}\right)^2\right]}, \tag{3.14}$$

$$\rho^{(u)^*} = \frac{2K_W\alpha_0^{(u)}}{\sum\limits_{k=1}^{K_W}\left[\left(1-\epsilon_k^{(\ell)}\right)\left(\ln m_k^{(1)}-\mu^{(u)^*}\right)^2+\epsilon_k^{(\ell)}\left(\ln m_k^{(2)}-\mu^{(u)^*}\right)^2\right]+K_W\left[2\beta_0^{(u)}+\left(\mu^{(u)^*}-\mu_0^{(u)}\right)^2\right]}, \tag{3.15}$$

where $\left(\rho^{(a)^*}, \mu^{(a)^*}, \rho^{(u)^*}, \mu^{(u)^*}\right)$ will be the updated distribution parameters in $\widehat{\boldsymbol{\Omega}}^{(\ell+1)}$.

The next step is to maximize $Q\left(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ with respect to $\boldsymbol{z}$ and $\boldsymbol{\eta}$. Note that this joint maximization is non-convex in general. Consider the following state-space model with parameters $(\boldsymbol{z}', \boldsymbol{\eta}')$ and binary observations $\boldsymbol{n}'$.

$$\begin{cases} n_k' \sim \text{Bernoulli}\left(\frac{1}{1+\exp(-z_k')}\right) \\ z_k' = c_0 z_{k-1}' + w_k' \\ w_k' \sim \mathcal{N}(0, \eta_k') \\ \eta_k' \sim \text{Inverse-Gamma}\left(a_0, b_0\right) \end{cases} \tag{3.16}$$

For the inference problem in (3.16), the log-posterior can be expressed as:

$$\underset{\boldsymbol{z}',\boldsymbol{\eta}'}{\arg\max}\ln \mathrm{P}\left(\boldsymbol{z}', \boldsymbol{\eta}' \mid \boldsymbol{n}'\right) = \underset{\boldsymbol{z}',\boldsymbol{\eta}'}{\arg\max}\left[\ln \mathrm{P}\left(\boldsymbol{\eta}' \mid \boldsymbol{n}'\right) + \mathrm{P}\left(\boldsymbol{z}' \mid \boldsymbol{\eta}', \boldsymbol{n}'\right)\right]. \tag{3.17}$$

If we replace the observations n'_k in (3.17) with $\epsilon_k^{(\ell)}$, for $k = 1, 2, \ldots, K_W$, the inference problem becomes equivalent to maximizing $Q\left(\boldsymbol{\Omega} \,\middle|\, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ in (3.11) with respect to $\boldsymbol{z}$ and $\boldsymbol{\eta}$.

In [16, 71], the inference of the parameters in (3.16) has been carried out through the EM algorithm, where in each iteration, a Kalman filtering and smoothing algorithm has been employed together with Gaussian approximations. Similar to [51], we refer to this EM algorithm as the inner EM not to confuse it with the EM algorithm we have already adopted, which we call the outer EM hereafter. The basic idea behind the inner EM is to approximate the solutions to (3.17) as:

$$\begin{cases} \boldsymbol{\eta'}^{*} = \arg\max_{\boldsymbol{\eta'}} \mathrm{P}\left(\boldsymbol{\eta'} \,\middle|\, \boldsymbol{n'}\right) \\ \\ \boldsymbol{z'}^{*} = \arg\max_{\boldsymbol{z'}} \mathrm{P}\left(\boldsymbol{z'} \,\middle|\, \boldsymbol{\eta'}^{*}, \boldsymbol{n'}\right) \end{cases}, \tag{3.18}$$

where $\boldsymbol{\eta'}^{*}$ are estimated through the inner EM with $\boldsymbol{z'}$ as the latent variables, and $\boldsymbol{z'}^{*}$ are just the result of a Kalman filtering and smoothing algorithm in (3.16) for $\boldsymbol{\eta'} = \boldsymbol{\eta'}^{*}$.

In order to make the inference procedure suitable for real-time implementation, we can avoid the inner EM and instead use crude estimates of $\boldsymbol{\eta'}^{*}$ in (3.18). Note that $\epsilon_k^{(\ell)}$, which acts as the observation n'_k in (3.16) for $k = 1, 2, \ldots, K_W$, is equal to $\mathrm{P}\left(n_k\!=\!1 \,\middle|\, m_k^{(1)}, m_k^{(2)}, \widehat{\boldsymbol{\Omega}}^{(\ell)}\right)$ calculated as in (3.10). Assuming that $\epsilon_k^{(\ell)} \approx \mathrm{P}\left(n'_k\!=\!1\right) = \frac{1}{1+\exp(-z'_k)}$, in the ℓ^{th} outer EM iteration, we can consider $\left[\mathrm{logit}\left(\epsilon_k^{(\ell)}\right) - c_0 \,\mathrm{logit}\left(\epsilon_{k-1}^{(\ell)}\right)\right]$ as a sample of $\mathcal{N}\left(0, {\eta'}_k\right)$. Therefore, considering the Inverse-Gamma prior, a crude estimate for ${\eta'}_k^{*}$ can be calculated for $k\!=\!1, 2, \ldots, K_W$ as:

$$\eta'^*_k = \frac{2b_0 + \left[\operatorname{logit}\left(\epsilon_k^{(\ell)}\right) - c_0 \operatorname{logit}\left(\epsilon_{k-1}^{(\ell)}\right)\right]^2}{2a_0 - 1}. \tag{3.19}$$

If K_W is small enough, we can simplify the state-space model of (3.16) by assuming a single variance, i.e., $\eta' = \eta'_k$ for $k = 1, 2, \ldots, K_W$, and using an estimate similar to (3.19) for η'^*. However, in this model, the crude estimate would be more reliable as it is based on K_W samples rather than a single sample. Considering a normalized log-likelihood and the same Inverse-Gamma prior on η', the estimate for η'^* can be computed as:

$$\eta'^* = \frac{2b_0 + \frac{1}{K_W}\sum_{k=1}^{K_W}\left[\operatorname{logit}\left(\epsilon_k^{(\ell)}\right) - c_0 \operatorname{logit}\left(\epsilon_{k-1}^{(\ell)}\right)\right]^2}{2a_0 - 1}. \tag{3.20}$$

After estimating η'^*_k in (3.19) for $k = 1, 2, \ldots, K_W$, or η'^* in (3.20), we can proceed as before to estimate $\boldsymbol{z'}^*$, i.e., using a Kalman filtering and smoothing algorithm with Gaussian approximations to estimate $\boldsymbol{z'}^*$ in (3.18). These estimates, namely $\boldsymbol{z}^*$ and $\boldsymbol{\eta}^*$, form approximate solutions for $\boldsymbol{z}$ and $\boldsymbol{\eta}$ in the original problem of maximizing $Q(\boldsymbol{\Omega} \mid \widehat{\boldsymbol{\Omega}}^{(\ell)})$ in (3.11) with respect to the state-space parameters.

Next, we discuss the details of the inner EM algorithm, as in [51], used to solve for $\boldsymbol{z}'$ and $\boldsymbol{\eta}'$ in (3.16). As mentioned before, the idea is to use an EM algorithm together with Gaussian approximations to maximize $\mathrm{P}\left(\boldsymbol{\eta}' \mid \boldsymbol{n}'\right)$, and then maximize the likelihood of $\boldsymbol{z'}$ with respect to the observations and estimated variances. Considering $\boldsymbol{z'}$ as the latent variables, the surrogate function $Q(\boldsymbol{\eta}' \mid \widehat{\boldsymbol{\eta}'}^{(\ell)})$ at ℓ^{th} EM iteration is calculated as:

$$Q\left(\boldsymbol{\eta}'\middle|\widehat{\boldsymbol{\eta}'}^{(\ell)}\right) = \mathbb{E}\left\{\ln \mathrm{P}\left(\boldsymbol{n}', \boldsymbol{z}' \mid \boldsymbol{\eta}'\right)\right\} + \ln \mathrm{P}(\boldsymbol{\eta}') \tag{3.21}$$
$$= \sum_{k=1}^{K_W} \left[\frac{\mathbb{E}\left\{(z'_k - c_0 z'_{k-1})^2\right\} + 2b_0}{2\eta'_k} + (a_0+1.5)\ln \eta'_k\right] + \mathsf{cnst.},$$

where the expectations are with respect to $\boldsymbol{z}' \mid \boldsymbol{n}', \widehat{\boldsymbol{\eta}'}^{(\ell)}$, and the $\mathsf{cnst.}$ term contains all the terms that are independent of $\boldsymbol{\eta}'$.

In the M-step of the inner EM algorithm, $Q\left(\boldsymbol{\eta}'\middle|\widehat{\boldsymbol{\eta}'}^{(\ell)}\right)$ is maximized with respect to $\boldsymbol{\eta}'$ to calculate the updated variances for the next EM iteration. Taking the derivative of (3.21) with respect to $\boldsymbol{\eta}'$ and equating it to zero results in the following update rule for $\widehat{\boldsymbol{\eta}'}^{(\ell+1)}$:

$$\widehat{\eta'}_k^{(\ell+1)} = \frac{1}{2a_0+3}\left[\mathbb{E}\left\{(z'_k - c_0 z'_{k-1})^2\right\} + 2b_0\right] \tag{3.22}$$
$$= \frac{1}{2a_0+3}\left[\mathbb{E}\left\{{z'_k}^2\right\} + c_0^2\mathbb{E}\left\{{z'_{k-1}}^2\right\} - 2c_0\mathbb{E}\left\{z'_k z'_{k-1}\right\} + 2b_0\right]$$
$$= \frac{1}{2a_0+3}\Big[\sigma^2_{k|K_W} + \bar{z}^2_{k|K_W} + c_0^2\sigma^2_{k-1|K_W} + c_0^2\bar{z}^2_{k-1|K_W} - 2c_0\sigma^2_{k,k-1|K_W}$$
$$-2c_0\bar{z}_{k|K_W}\bar{z}_{k-1|K_W} + 2b_0\Big],$$

where the parameters $\bar{z}_{k|K_W}$ and $\sigma^2_{k|K_W}$ in Eq. (3.22) are respectively the mean and the variance of $z'_k \mid \boldsymbol{n}', \widehat{\boldsymbol{\eta}'}^{(\ell)}$.

If we consider the Gaussian approximation $\mathcal{N}\left(\bar{z}_{k_1|k_2}, \sigma^2_{k_1|k_2}\right)$ to the density $z'_{k_1} \mid n'_{1:k_2}, \widehat{\boldsymbol{\eta}'}^{(\ell)}$ for $1 \le k_1 \le k_2 \le K_W$, these parameters can be computed in a forward and backward pass similar to the conventional Kalman filtering and smoothing algorithms. The corresponding filtering equations for $1 \le k \le K_W$ are summarized as:

$$\begin{cases} \bar{z}_{k|k-1} = c_0 \bar{z}_{k-1|k-1} \\ \sigma^2_{k|k-1} = c_0^2 \sigma^2_{k-1|k-1} + {\eta'_k}^{(l)} \\ \bar{z}_{k|k} = \bar{z}_{k|k-1} + \sigma^2_{k|k-1} \left[n'_k - \frac{\exp(\bar{z}_{k|k})}{1+\exp(\bar{z}_{k|k})} \right] \\ \sigma^2_{k|k} = \left[\frac{1}{\sigma^2_{k|k-1}} + \frac{\exp(\bar{z}_{k|k})}{\left(1+\exp(\bar{z}_{k|k})\right)^2} \right]^{-1} \end{cases} \tag{3.23}$$

Note that the third equation in (3.23) is a non-linear equation whose solution can be approximated through standard approaches such as the Newton's method. The last two equations in (3.23) come from the Gaussian approximation: assuming that $z'_{k-1} \,|\, n'_{1:k-1}, \widehat{\boldsymbol{\eta}'}^{(\ell)} \backsim \mathcal{N}\left(\bar{z}_{k-1|k-1}, \sigma^2_{k-1|k-1}\right)$ we calculate the Gaussian approximation for $z'_k \mid n'_{1:k}, \widehat{\boldsymbol{\eta}'}^{(\ell)}$. The mean of the Gaussian approximation $\bar{z}_{k|k}$ is calculated as the mode of $\ln \mathrm{P}\left(z'_k \mid n'_{1:k}, \widehat{\boldsymbol{\eta}'}^{(\ell)}\right)$, and its variance $\sigma^2_{k|k}$ is computed as the negative inverse Hessian of $\ln \mathrm{P}\left(z'_k \mid n'_{1:k}, \widehat{\boldsymbol{\eta}'}^{(\ell)}\right)$ evaluated at the estimated mean $\bar{z}_{k|k}$ [72]. The smoothing equations are the same as those used for fixed interval smoothing. Therefore, for $1 \leq k \leq K_W - 1$, we have:

$$\begin{cases} s_k = \sigma^2_{k|k} \Big/ \sigma^2_{k+1|k} \\ \bar{z}_{k|K_W} = \bar{z}_{k|k} + s_k \left(\bar{z}_{k+1|K_W} - \bar{z}_{k+1|k}\right) \\ \sigma^2_{k|K_W} = \sigma^2_{k|k} + s_k^2 \left(\sigma^2_{k+1|K_W} - \sigma^2_{k+1|k}\right) \end{cases} \tag{3.24}$$

The $\sigma^2_{k,k-1|K_W}$ term in (3.22) is a lagged covariance term that can be computed using the covariance smoothing algorithm [73]:

$$\sigma^2_{k,k-1|K_W} = \text{Cov}\left\{ z'_k, z'_{k-1} \mid \boldsymbol{n'}, \widehat{\boldsymbol{\eta'}}^{(\ell)} \right\} = \frac{\sigma^2_{k-1|k-1}\sigma^2_{k|K_W}}{\sigma^2_{k|k-1}}. \tag{3.25}$$

Having calculated the variances $\boldsymbol{\eta'}^*$ from the inner EM algorithm, $\boldsymbol{z'}^*$ can be estimated using a single forward and backward pass for $\boldsymbol{\eta'} = \boldsymbol{\eta'}^*$, similar to that used in the inner EM algorithm. In summary, we have transformed the problem of maximizing (3.11) with respect to $\boldsymbol{z}$ and $\boldsymbol{\eta}$ into inferring $\boldsymbol{z'}$ and $\boldsymbol{\eta'}$ in (3.16) by identifying n'_k with $\epsilon^{(l)}_k$ for $k = 1, \dots, K_W$. We have then solved the latter problem through an EM algorithm combined with Gaussian approximations and Kalman filtering and smoothing. Therefore, we have $\boldsymbol{z}^* = \boldsymbol{z'}^*$ and $\boldsymbol{\eta}^* = \boldsymbol{\eta'}^*$ in the original problem.

Algorithm 6 summarizes the overall inference procedure within a fixed-lag window of length K_W. Going back to Fig. 3.2, copied from the paper, we assume $k = k_0$ is the current instance and the goal is to infer the attentional state at instance $k = k_0 - K_F$ based on the attention markers within the window indexed from 1 to K_W, given by $m^{(i)}_k$ for $i = 1, 2$ and $k = 1, \dots, K_W$. We initialize the state-space model parameter set $\boldsymbol{\Omega}$ using the estimates at the previous instance, and the output of Algorithm 6, i.e., $\widehat{\boldsymbol{\Omega}}$, is used for initialization in the next instance. Defining $f(.)$ as the sigmoid function, $f\left(\hat{z}_{K_W-K_F}\right)$ determines the estimated probability of attending to speaker 1 at $k = k_0 - K_F$, and $\left[f\left(\hat{z}_{K_W-K_F} - 1.65\hat{\sigma}^2_{K_W-K_F|K_W}\right), f\left(\hat{z}_{K_W-K_F} + 1.65\hat{\sigma}^2_{K_W-K_F|K_W}\right)\right]$ represents the %90 confidence intervals of this estimate, where $\hat{\sigma}^2_{K_W-K_F|K_W}$ represents the inferred variance of $\hat{z}_{K_W-K_F}$ calculated through the discussed Gaussian approximations. The parameter $\mathcal{S}_{EM}$ in Algorithm 6 is a stopping condition for the outer EM, which can be a limit on the number of iterations.

Algorithm 6 Parameter Estimation in Dynamic State-Space Model

```
Inputs: $m^{(1)}_{1:K_W}$, $m^{(2)}_{1:K_W}$, $\alpha^{(a)}_0$, $\alpha^{(u)}_0$, $\beta^{(a)}_0$, $\beta^{(u)}_0$, $\mu^{(a)}_0$, $\mu^{(u)}_0$, $a_0$, $b_0$, $\mathcal{S}_{EM}$
Output: $\widehat{\mathbf{\Omega}} = \{\hat{z}_{1:K_W}, \hat{\eta}_{1:K_W}, \hat{\rho}^{(a)}, \hat{\mu}^{(a)}, \hat{\rho}^{(u)}, \hat{\mu}^{(u)}\}$
Set $\widehat{\mathbf{\Omega}}^{(0)}$ as the initialization for state-space model parameter set based on estimates in the previous instance
$\ell = 0$
while $\neg\mathcal{S}_{EM}$ do
    calculate $\epsilon^{(\ell)}_{1:K_W}$ using (3.10)
    update the parameters of the Log-Normal distributions, i.e., $\mu^{(a)}$, $\mu^{(u)}$, $\rho^{(a)}$, $\rho^{(u)}$, based on equations (3.12), (3.13), (3.14), and (3.15) respectively
    update the state-space variances, i.e., $\eta_{1:K_W}$, using the inner-EM algorithm or the crude estimates in equations (3.19) and (3.20)
    update the hidden states in the state-space model, i.e., $z_{1:K_W}$, using a Kalman filtering and smoothing algorithm with Gaussian approximations
    set $\widehat{\mathbf{\Omega}}^{(\ell+1)}$ as the updated parameter set including the updated distribution parameters, variances, and hidden states in the state-space model
    $\ell \leftarrow \ell + 1$
end while
$\widehat{\mathbf{\Omega}} = \widehat{\mathbf{\Omega}}^{(\ell)}$.
```

3.1.6 EEG Recording and Experiment Specifications

64-channel EEG was recorded using the actiCHamp system (Brain Vision LLC, Morrisville, NC, US) and active EEG electrodes with Cz channel being the reference. The data was digitized at a 10 KHz sampling frequency. Insert earphones ER-2 (Etymotic Research Inc., Elk Grove Village, IL, US) were used to deliver sound to the subjects while sitting in a sound-attenuated booth. The earphones were driven by the clinical audiometer Piano (Inventis SRL, Padova, Italy), and the volume was adjusted for every subject's right and left ears separately until the loudness in both ears was matched at a comfortably loud listening level. Three normal-hearing adults participated in the study. The mean age of subjects was 49.5 years with the standard deviation of 7.18 years. The study included a constant-attention experiment, where the subjects were asked to sit in front of a computer screen and restrict motion while any audio was playing. The data used in this chapter corresponds to 3 subjects, 24 trials each.

The stimulus set contained eight story segments, each approximately ten minutes long. Four segments were narrated by male speaker 1 (M1) and the other four by male speaker 2 (M2). The stimuli were presented to the subjects in a dichotic fashion, where various stories read by M1 were played in the left ear, while stories read by M2 were played in the right ear for all the subjects. Each subject listened to twenty four trials of the dichotic stimulus. Each trial had a duration of approximately one minute, and for each subject, no storyline was repeated in more than one trial. During each trial, the participants were instructed to look at an arrow at the center of the screen, which determined whether to attend to the right-ear story or to the left one. The arrow remained fixed for the duration of each trial, making it a constant-attention experiment. At the end of each trial, two multiple choice semantic questions about the attended story were displayed on the screen to keep the subjects alert. The responses of the subjects as well as their reaction time were recorded as a behavioral measure of the subjects' level of attention, and above eighty percent of the questions were answered correctly by each subject. Breaks and snacks were given between stories if requested. All the audio recordings, corresponding questions, and transcripts were obtained from a collection of stories recorded at

Hafter Auditory Perception Lab at UC Berkeley.

3.1.7 MEG Recording and Experiment Specifications

MEG signals were recorded with a sampling rate of 1 KHz using a 160-channel whole-head system (Kanazawa Institute of Technology, Kanazawa, Japan) in a dimly lit magnetically shielded room (Yokogawa Electric Corporation). Detection coils were arranged in a uniform array on a helmet-shaped surface on the bottom of the dewar with 25 mm between the centers of two adjacent 15.5 mm diameter coils. Also, sensors are set as first-order axial gradiometers with a baseline of 50 mm, resulting in field sensitivities of $5\,\frac{\mathrm{fT}}{\sqrt{\mathrm{Hz}}}$ or better in the white noise region.

The two speech signals had approximately 65 dB SPL and were presented using the software package Presentation (Neurobehavioral Systems Inc., Berkeley, CA, US). The stimuli were delivered to the subjects' $\tilde{O}$ ears with $50\,\Omega$ sound tubing (E-A-RTONE 3A; Etymotic Research), attached to E-A-RLINK foam plugs inserted into the ear canal. Also, the whole acoustic delivery system was equalized to give an approximately flat transfer function from 40 Hz to 3000 Hz. A 200 Hz low-pass filter and a notch filter at 60 Hz were applied to the magnetic signal in an online fashion for noise removal. Three of the 160 channels were magnetometers separated from the others and used as reference channels. Finally, to quantify the head movement, five electromagnetic coils were used to measure each subject's head position inside the MEG machine once before and once after the experiment.

Nine normal-hearing, right-handed young adults (ages between 20 and 31) participated in this study. The study includes two sets of experiments: the constant-attention experiment and the attention-switch experiment, in both of which six subjects participated. Three subjects took part in both of the experiments. The experimental procedure were approved by the University of Maryland Institutional Review Board (IRB), and written informed consent was obtained from each subject before the experiment.

The stimuli included four non-overlapping segments from the book *A Child's History of England* by Charles Dickens. Two of the segments were narrated by a man and the other two by a woman. Three different mixtures, each 60 s long, were generated and used in the experiments to prevent reduction in the attentional focus of the subjects. Each mixture included a segment narrated by the male speaker and one narrated the the female speaker. In all trials, the stimuli were delivered diotically to both ears using tube phones inserted into the ear canals at a roughly 65 dB SPL, as mentioned. The constant-attention experiment consisted of two conditions: 1) attending to the male speaker in the first mixture, 2) attending to the female speaker in the second mixture. In the attention-switch experiment, subjects were instructed to focus on the female speaker in the first 28 s of the trial, switch their attention to the male speaker after hearing a 2 s pause (28th to 30th seconds), and maintain their focus on the latter speaker through the end of the trial. Each mixture was repeated three times in the experiments, resulting in six trials per speaker for the constant-attention experiment and three trials per speaker for the attention-switch experiment. After the presentation of each mixture, subjects answered comprehensive questions related to the segment they were instructed to focused on, as a way to keep them motivated on attending to the target speaker. Eighty percent of the questions were answered correctly on average. Furthermore, a pilot study for each of the nine participating subjects was performed prior to the main experiments. In this study, the subjects listened to a single speech stream, first segment in the stimuli set narrated by the male speaker, for three trials each 60 s long. The MEG recordings in the pilot study were used to calculate the subject-specific linear combination of MEG channels which forms the auditory component of the response, as will be explained next. Note that for each subject, all the recordings were performed in a single session resulting in a minimal change of the subject's head position with respect to the MEG sensors.

3.2 Results

In this section, we apply our real-time attention decoding framework to synthetic data as well as M/EEG recordings. In order to validate our proposed framework, we perform two sets of simulations. The simulation in Section 3.2.1 pertains to our EEG analysis and employs a decoding model, while the simulation in Section 3.2.2 corresponds to our MEG analysis and uses an encoding model. The results for the analysis of EEG and MEG recordings are demonstrated in Section 3.2.3 and 3.2.4, respectively.

3.2.1 Decoding Model Simulation

3.2.1.1 Simulation Settings

In order to simulate EEG data under a dual-speaker condition, we use the following generative model:

$$e_t = w_t^{(1)}\Big(s_t^{(1)} * h_t\Big) + w_t^{(2)}\Big(s_t^{(2)} * h_t\Big) + \mu + n_t \tag{3.26}$$

where $s_t^{(1)}$ and $s_t^{(2)}$ are respectively the speech envelopes of speakers 1 and 2 at time t; the output e_t is the neural response, which denotes an auditory component of the EEG recordings or the measured EEG response at a given channel at time t for $t = 1, 2, \ldots, T$. Motivated by the analysis of LTI systems, h_t can be considered as the impulse response of the neural process resulting in e_t, and $*$ represents the convolution operator; the scalar μ is an unknown constant mean, and n_t denotes a zero-mean i.i.d Gaussian noise. The weight functions $w_t^{(1)}$ and $w_t^{(2)}$ are signals modulated by the attentional state which determine the contributions of speakers 1 and 2 to e_t, respectively. In order to simulate the attention modulation effect, we assume that when speaker 1 (resp. 2) is attended to at time t, we have $w_t^{(1)} > w_t^{(2)}$ (resp. $w_t^{(1)} < w_t^{(2)}$).

We have chosen two 60 s-long speech segments from those used in the MEG experiment (See section 3.1.7) and calculated $s_t^{(1)}$ and $s_t^{(2)}$ as their envelopes for a sampling rate of $f_s = 200\,\text{Hz}$. Also, we have set $\mu = 0.02$ and $n_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 2.5{\times}10^{-5})$ in Eq. (3.26). Fig. 3.3-A shows the location and amplitude of the lag components in the impulse response, which is then smoothed using a Gaussian kernel with standard deviation of 10 ms to result in the final impulse response h_t, shown in Fig. 3.3–B. The significant components of h_t are chosen at 50 ms and 100 ms lags, with few smaller components at higher latencies [51]. The weight signals $w_t^{(1)}$ and $w_t^{(2)}$ in Eq. (3.26) are chosen to favor speaker 1 in the $[0\,\mathrm{s}, 30\,\mathrm{s})$ interval and speaker 2 in the $(30\,\mathrm{s}, 60\,\mathrm{s}]$ interval, with the transition happening within a 3 s interval around the 30 s mark.

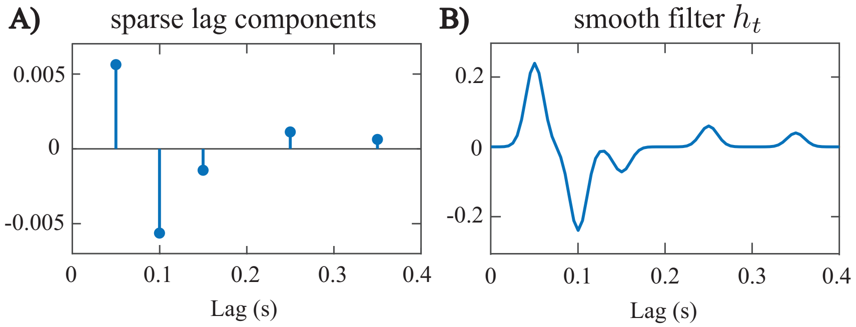

Figure 3.3: Impulse response h_t used in Eq. (3.26). A) sparse lag components, B) the smooth impulse response.

3.2.1.2 Parameter Selection

We aim at estimating decoders in this simulation, which linearly map $\mathbf{e}_t$ and its lags to $s_t^{(1)}$ and $s_t^{(2)}$. To estimate the decoders, we have considered consecutive non-overlapping windows of length 0.25 s resulting in $K = 240$ windows of length $W = 50$ samples. Also, we have chosen $\gamma = 0.001$ through cross-validation and $\lambda = 0.95$ in estimating the decoding coefficients, which results in an *effective* data length of 5 s for decoder estimation. The forward lags of the neural response have been limited to a 0.4 s window, i.e., $L_d = 80$ samples. Given that the decoder corresponds to the inverse of a smooth kernel h_t, it may not have the same smoothness properties of h_t. Hence, we do not employ a smooth basis for decoder estimation. We have used the FASTA package [69] with Nesterov's acceleration method to implement the forward-backward splitting algorithm for encoder/decoder estimation. As for the state-space model estimators, we have considered 20 (inner and outer) EM iterations for the batch-mode estimates that use the entire data, while for the real-time estimates, we use 1 inner EM iteration and 20 outer EM iterations (See Section 3.1.5 for more details).

There are three criteria for choosing the fixed-lag smoothing parameters: First, how close to the true real-time analysis the system operates is determined by K_F. Second, the computational cost of the system is determined by K_W. Third, how close the output of the system is to that of batch-mode processing is determined by both K_F and K_W. These three criteria form a tradeoff in tuning the parameters K_W and K_F. Specific choices of these parameters are given in the next subsection.

For tuning the hyperparameters of the priors on the attended and unattended distributions, we have used a separate 15 s sample trial generated from the same simulation model in Eq. (3.26) for each of the three cases. The parameters $\left(\alpha_0^{(a)}, \alpha_0^{(u)}, \beta_0^{(a)}, \beta_0^{(u)}, \mu_0^{(a)}, \mu_0^{(u)}\right)$ have been chosen by fitting the Log-Normal distributions to the attention marker outputs from the sample trials in a supervised manner (with known attentional state). The variance of the Gamma priors $\frac{\alpha_0^{(a)}}{\beta_0^{(a)2}}$ and $\frac{\alpha_0^{(u)}}{\beta_0^{(u)2}}$ have been chosen large enough such that the priors are non-informative. This step can be thought of as the initialization of the algorithms prior to data analysis. For the Inverse-Gamma prior on the state-space variances, we have chosen $a_0 = 2.008$ and $b_0 = 0.2016$, resulting in a mean of 0.2 and a variance of 5. This prior favors small values of η_k's to ensure that the state estimates are immune to large fluctuations of the attention markers, while the large variance (compared to the mean) results in a non-informative prior.

3.2.1.3 Estimation Results

Fig. 3.4 shows the results of our estimation framework for a correlation-based attention marker. Row A in Fig. 3.4 shows three cases considered for modulating the weights $w_t^{(1)}$ and $w_t^{(2)}$, where the weights are contaminated with Gaussian noise $\mathcal{N}(0, 4\times10^{-4})$. Cases 1, 2, and 3 exhibit increasing levels of difficulty in discriminating the contributions of the two speakers to the neural response. Rows B and C in Fig. 3.4 respectively show the decoder estimates for speakers 1 and 2. As expected, the significant components of the decoders around 50 ms, 100 ms, and 150 ms lags, are modulated by the attentional state, and the modulation effect weakens as we move from Case 1 to 3. In Case 1, these components are less significant overall for the decoder estimates of speaker 2 in the $[0\,\text{s}, 30\,\text{s}]$ time interval and become larger as the attention switches to speaker 2 during the rest of the trial (red boxes in row C of Case 1). On the other hand, in Case 3, the magnitude of the said components do not change notably across the 30 s mark.

We have considered two different attention markers for this simulation. Row D in Fig. 3.4 displays the output of a correlation-based attention marker for speakers

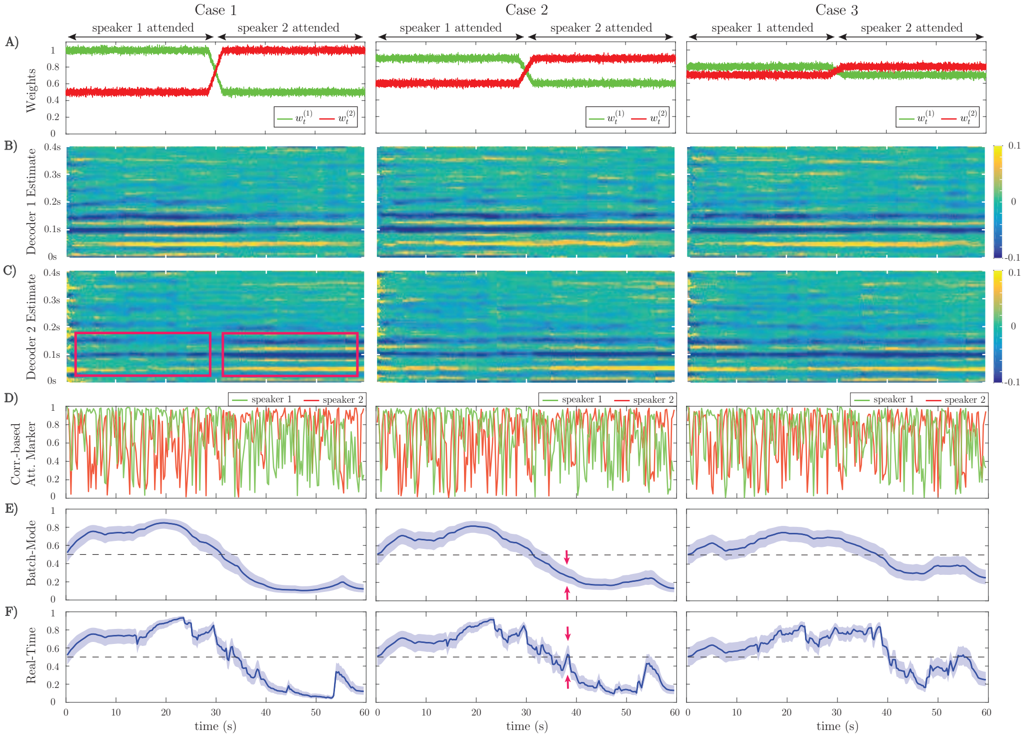

Figure 3.4: Estimation results of application to simulated EEG data for the correlation-based attention marker: A) Input weights $w_t^{(1)}$ and $w_t^{(2)}$ in Eq. (3.26), which determine the relative effect of the two speeches on the neural response. Based on our generative model, the attention is on speaker 1 for the first half of each trial and on speaker 2 for the second half. Case 1 corresponds to a scenario where the effects of the attended and unattended speeches in the neural response are well-separated. This separation decreases as we move from Case 1 to Case 3. B) Estimated decoder for speaker 1. C) Estimated decoder for speaker 2. In Case 1, the significant components of the estimated decoders near the 50 ms, 100 ms, and 150 ms lags are notably modulated by the attentional state as highlighted by the red boxes. This effect weakens in Case 2 and visually disappears in Case 3. D) Output of the correlation-based attention marker for each speaker. E) Output of the batch-mode state-space estimator for the correlation-based attention marker as the estimated probability of attending to speaker 1. F) Output of the real-time state-space estimator, i.e., fixed-lag smoother, for the correlation-based attention marker as the estimated probability of attending to speaker 1. The real-time estimator is not as robust as the batch-mode estimator to the stochastic fluctuations of the attention marker in row D and is more prone to misclassifications. The red arrows in rows E and F of Case 2 show that the batch-mode estimator correctly classifies the instance as attending to speaker 2, while the real-time estimator is unable to determine the attentional state.

1 and 2, which is calculated as $m_k^{(i)} = \left|\text{corr}\left(\mathbf{y}_k^{(i)}, \mathbf{X}_k \hat{\boldsymbol{\theta}}_k^{(i)}\right)\right|$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$. As discussed in subsection 3.1.3, this attention marker is a measure of how well a decoder can reconstruct its target envelope. As observed in row D of Fig. 3.4, the attention marker is a highly variable surrogate of the attentional state at each instance, i.e., *on average* the attention marker output for speaker 1 is higher then that of speaker 2 in the $[0\,\text{s}, 30\,\text{s})$ interval and vice versa in the $(30\,\text{s}, 60\,\text{s}]$ interval. The reliability of the attention marker significantly degrades going from Case 1 to 3. This highlights the need for state-space modeling and estimation in order to optimally exploit the attention marker.

Rows E and F in Fig. 3.4 respectively show the batch-mode and real-time estimates of the attentional state probabilities $p_k = \text{P}\left(n_k = 1\right)$ for $k = 1, \ldots, K$, for the correlation-based attention marker, where colored halls indicate 90% confidence intervals. Row F in Fig. 3.4 corresponds to the fixed-lag smoother, using a window of length 15 s ($K_W = \lfloor 15 f_s / W \rfloor$), and a forward-lag of 1.5 s ($K_F = \lfloor 1.5 f_s / W \rfloor$). We refer to this estimator as the real-time estimator henceforth. Note that by accounting for the forward-lag in the decoder (L_d), the overall delay in estimating the attentional state is 1.9 s. Recall that in batch-mode processing, all of the attention marker outputs across the trial are available the state-space estimator, as opposed to the fixed-lag estimator which has access to a limited number of the attention markers. Therefore, the output of the batch-mode estimator (Row E) is a more robust measure of the instantaneous attentional state as compared to the real-time estimator (Row F), since it is less sensitive to the stochastic fluctuations of the attention markers in row D. For example, in the instance marked by the red arrows in rows E and F of Case 2 in Fig. 3.4, the batch-mode estimator classifies the instance correctly as attending to speaker 2, while the real-time estimator cannot make an informed decision since $p_k = 0.5$ falls within the 90% confidence interval of the estimate at this instance. However, the real-time estimator exhibits performance closely matching that of the batch-mode estimator for most instances, while operating in real-time with limited data access and significantly lower computational complexity.

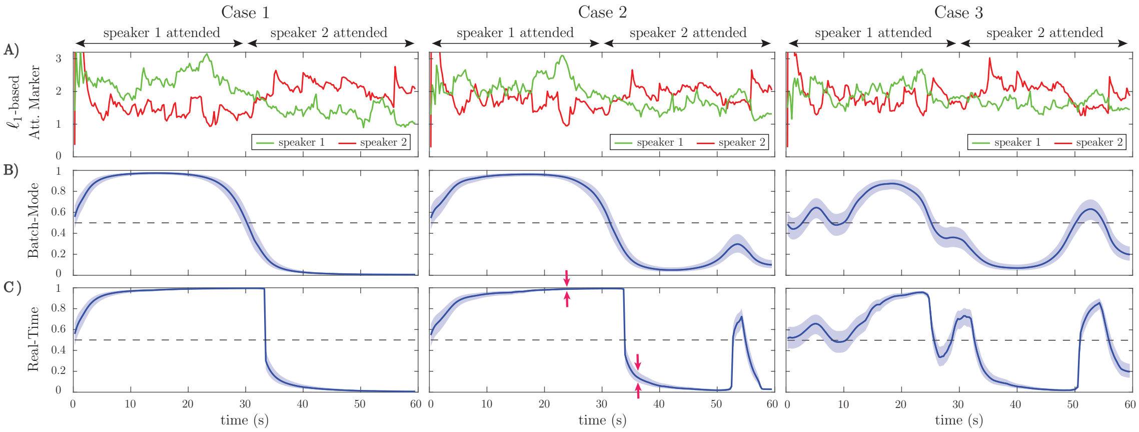

Figure 3.5: Estimation results of application to simulated EEG data for the ℓ_1-based attention marker: A) Output of the ℓ_1-based attention marker for each speaker, corresponding to the three cases in Figure 3.4. B) Output of the batch-mode state-space estimator for the ℓ_1-based attention marker as the estimated probability of attending to speaker 1. C) Output of the real-time state-space estimator for the ℓ_1-based attention marker as the estimated probability of attending to speaker 1. Similar to the preceding correlation-based attention marker, the classification performance degrades when moving from Case 1 (strong attention modulation) to Case 3 (weak attention modulation).

Row A in Fig. 3.5 exhibits the output of another attention marker computed as the ℓ_1-norm of the decoder given by $m_k^{(i)} := \left\| \hat{\boldsymbol{\theta}}_k^{(i)} \right\|_1$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$, where the first element of $\hat{\boldsymbol{\theta}}_k^{(i)} \in \mathbb{R}^{L_d+2}$ (the intercept parameter) is discarded in computing the ℓ_1 norm. This attention marker captures the effect of the significant peaks in the decoder. The rationale behind using the ℓ_1-norm based attention marker is the following: in the extreme case that the neural response is solely driven by the attended speech, we expect the unattended decoder coefficients to be small in magnitude and randomly distributed across the time lags. The attended decoder, however, is expected to have a sparse set of informative and significant components corresponding to the specific latencies involved in auditory processing. Thus, the ℓ_1 norm serves to distinguish between these two cases. Rows B and C in Fig. 3.5 show the batch-mode and real-time estimates of the attentional state probabilities for the ℓ_1-norm attention marker, respectively, where colored halls indicate 90% confidence intervals. Consistent with the results of the correlation-based attention marker (Rows E and F in Fig. 3.4), the real-time estimator exhibits performance close to that of the batch-mode estimator. Comparing Figs. 3.4 and 3.5 reveals the dependence of the attentional state estimation performance on the choice of the attention marker: while the correlation-based attention marker is more widely used, the ℓ_1-based attention marker provides smoother estimates of the attention probabilities, and can be used as a more robust alternative to the correlation-based attention marker.

3.2.1.4 Discussion and Further Analysis

Going from Case 1 to Case 3 in Fig. 3.4 and Fig. 3.5, we observe that the performance of all estimators degrades, causing a drop in the classification accuracy and confidence. This performance degradation is due to the declining power of the attention markers in separating the contributions of the attended and unattended speakers. However, comparing the outputs of the real-time and batch-mode estimators with their corresponding attention marker outputs in row D of Fig. 3.4 and row A of Fig. 3.5, highlights the role of the state-space model in suppressing the stochastic fluctuations of the attention markers and thereby providing a robust and smooth measure of the attentional state.

It is noteworthy that all the estimators exhibit a systematic delay in detecting the deflection point at 30 s, even for the well-separated Case 1 and batch-mode estimation. This delay is due to two main factors: first, the transition period of 3 s in the design of the weight signals contributes to this delay. Second, although the forgetting factor mechanism used in estimating the decoder coefficients results in more stable estimates, it causes an extra delay to the overall performance of the estimator.

Comparing the batch-mode and the real-time estimators in Fig. 3.4 and Fig. 3.5, we observe that the real-time estimators closely follow the output of the batch-mode estimators, while having access to data in an online fashion. A significant deviation between the batch-mode and real-time performance is observed in rows B and C (Cases 1 and 2) of Fig. 3.5 in the form of sharp drops in the real-time estimates of the attentional state probability. Given that the real-time estimator has only access to the attention marker within K_F samples in the future, the confidence intervals significantly narrow down within the first half of the trial, as all the past and near-future observations are consistent with attention to speaker 1. However, shortly after the 30 s mark the estimator detects the change and the confidence bounds widen accordingly (see red arrows in row C of Case 2 in Fig. 3.5).

In order to further quantify the performance gap between the batch-mode and real-time estimators, we define their relative Mean Squared Error (MSE) as:

$$\mathrm{MSE} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{1+\exp\left(-\hat{z}_k^{(B)}\right)} - \frac{1}{1+\exp\left(-\hat{z}_k^{(R)}\right)}\right)^2 \tag{3.27}$$

where $\hat{z}_{1:K}^{(R)}$ and $\hat{z}_{1:K}^{(B)}$ denote the real-time and batch-mode state estimates over a given trial, respectively. We have considered the logistic transformation of $\hat{z}_{1:K}^{(B)}$ and $\hat{z}_{1:K}^{(R)}$, which gives the probability of attending to speaker 1.

Figure 3.6 shows the effect of varying the forward-lag K_F from 0 s (i.e., fully real-time) to 5 s with 0.5 s increments for the two attention markers in Case 2 of Fig. 3.4 and Fig. 3.5, as an example. All of the other parameters in the simulation have been fixed as before. The left panels in Fig. 3.6 show the MSE for different values of K_F in the real-time setting. As expected, for both attention markers, the MSE decreases as the forward-lag increases. The right panels in Fig. 3.6 display the incremental MSE defined as the change in MSE when K_F is increased by 0.5 s, starting from $K_F = 0$ s. Notice that even a 0.5 s forward-lag significantly decreases the MSE from $K_F = 0$ s. The subsequent improvements of the MSE diminish as K_F is increased further. Our choice of $K_F = 1.5$ s in the foregoing analysis was made to maintain a reasonable tradeoff between the MSE improvement and the delay in real-time operation.

Finally, Fig. 3.7 shows the estimated attention probabilities and their 90% confidence intervals for the correlation-based attention marker in Case 2 of Fig. 3.4,

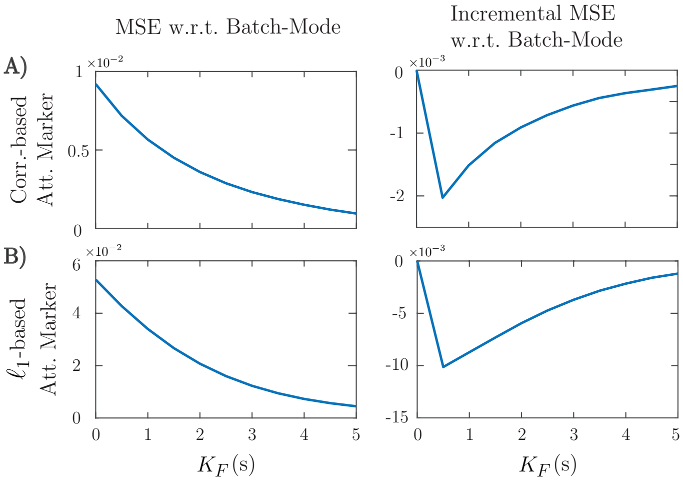

Figure 3.6: Effect of the forward-lag K_F on the MSE for the two attention markers in case 2 of Fig. 3.4 and Fig. 3.5. A) Correlation-based attention marker, B) ℓ_1-based attention marker. As the forward-lag increases, the MSE decreases, and the output of the real-time estimator becomes more similar to that of the batch-mode. This results in more robustness for the real-time estimator at the expense of more delay in decoding the attentional state. The right panels show that the incremental improvement to the MSE decreases as K_F increases.

as an example. The three curves correspond to the extreme values of K_F in Fig. 3.6 given by $K_F = 0\,\mathrm{s}$ (blue) and $K_F = 5\,\mathrm{s}$ (red), and the batch-mode estimate (green). All the other parameters have been fixed as explained before. The fixed-lag smoothing approach with $K_F = 5\,\mathrm{s}$ is as robust as the batch-mode estimate. The fully real-time estimate with $K_F = 0\,\mathrm{s}$ follows the same trend as the other two. However, it is susceptible to the stochastic fluctuations of attention marker, which may lead to misclassifications (see the red arrows in Fig. 3.7).

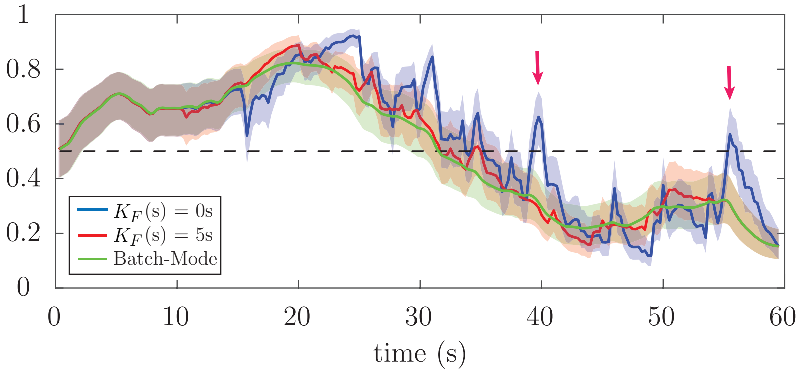

Figure 3.7: Estimated attention probabilities together with their 90% confidence intervals for the correlation-based attention marker in Case 2 of Fig. 3.4. The blue, red and green curves correspond to $K_F = 0\,\mathrm{s}$, $K_F = 5\,\mathrm{s}$, and batch-mode estimation, respectively. The estimator for $K_F = 5\,\mathrm{s}$ is nearly as robust as the batch-mode. However, the fully real-time estimator with $K_F = 0\,\mathrm{s}$ is sensitive to the stochastic fluctuations of the attention markers, which results in the misclassification of the attentional state at the instances marked by red arrows.

3.2.2 Encoding Model Simulation

3.2.2.1 Simulation Settings

Consider the following generative model to simulate MEG data under a dual-speaker condition:

$$e_t = s_t^{(1)} * \tau_t^{(1)} + s_t^{(2)} * \tau_t^{(2)} + \mu + n_t, \tag{3.28}$$

where e_t, $s_t^{(1)}$, and $s_t^{(2)}$ respectively denote the auditory component of the neural response, speech envelope for speaker 1, and speech envelope for speaker 2. We have used the same speech signals for $s_t^{(1)}$ and $s_t^{(2)}$ as in the EEG simulation, with the same sampling rate of $f_s = 200$ Hz. In the context of MEG processing, $\tau_t^{(1)}$ and $\tau_t^{(2)}$ are referred to as the TRF for speakers 1 and 2. We have set $\mu = 0.001$ as the unknown constant mean and $n_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 2.5 \times 10^{-7})$ as the observation noise. We assume an attention modulation effect on the M100 component of the TRFs.

Figure 3.8 shows two cases for the TRFs $\tau_t^{(1)}$ and $\tau_t^{(2)}$: In the left panels (case 1), there is a strong attention modulation effect on the M100 components, and in the right panels (case 2), this effect is weakened. In both cases, the attention is on speaker 1 during the $[0, 30)$ s interval and on speaker 2 during the $(30, 60]$ s interval. Also, we have considered a length of 0.4 s for the TRFs. Row B in Fig. 3.8 shows examples of the attended and the unattended TRFs for each of the two cases. In case 1, there is a large difference between the magnitude of the M100 components in the attended and the unattended TRFs, while in case 2, this difference is small compared to our estimation accuracy. We have also considered three higher latency components in the TRFs which are not modulated by the attentional state, similar to the M50 component. As shown in row A of Fig. 3.8, a zero-mean Gaussian i.i.d. noise is added to the TRF components as well. Note that similar to the EEG simulation, we have used a Gaussian kernel with the standard deviation of 10 ms to smooth the TRFs. This smoothness property is also observed in TRFs estimated from experimentally-recorded MEG signals [48, 49].

3.2.2.2 Parameter Selection

For the encoder estimation parameters in Algorithm 5, we have considered consecutive non-overlapping windows of length 0.25 s, i.e., $W = 50$, resulting in $K = 240$ instances, and we have assumed the same 0.4 s length for the TRFs, i.e., $L_e = 80$.

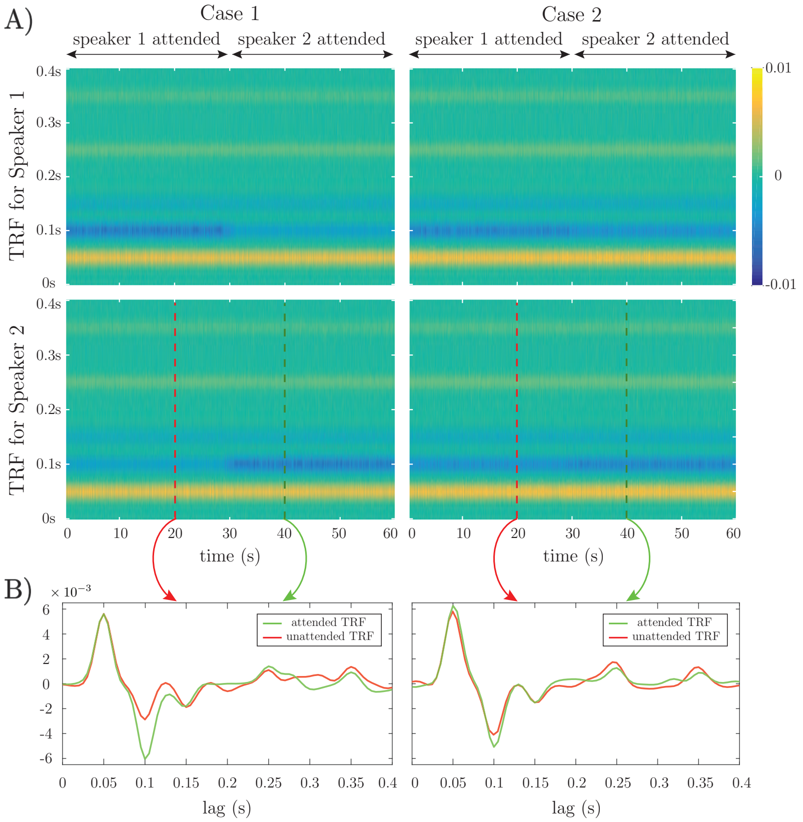

Figure 3.8: The TRFs $\tau_t^{(1)}$ and $\tau_t^{(2)}$ used for the simulation model in Eq. (3.28). A) TRFs for case 1 (strong modulation in M100 components) and case 2 (weak modulation in M100 components). B) Snapshots of the attended and unattended TRFs for the two cases.

We have chosen $\gamma = 0.005$ through cross-validation and $\lambda = 0.9167$, which results in an *effective* window length of 3 s for encoder estimation. Considering the smoothing Gaussian kernel used in the forward model, we have used the Gaussian dictionary matrix $\mathbf{G}_0 \in \mathbb{R}^{(L_e+1)\times(L_e+1)}$ for each speaker in the encoder estimation step to enforce smoothness in the TRFs. The dictionary columns consist of overlapping Gaussian kernels with the standard deviation of 10 ms, whose means cover the 0 s to 0.4 s lag with $T_s = 5$ ms increments. As a result, considering the simultaneous estimation of the two TRFs, the overall dictionary matrix would be $\mathbf{G} = \operatorname{diag}(1, \mathbf{G_0}, \mathbf{G_0})$.

We have used the FASTA package [68] with Nesterov's acceleration method to implement the forward-backward splitting algorithm. All the prior distribution parameters of the state-space models are set similar to the EEG simulation in the paper, where $a_0 = 2.008$, $b_0 = 0.2016$, and the prior parameters for the attended and unattended distributions were tuned based on a separate 15 s sample trial. For the real-time state-space estimator, we have used a sliding window of length 15 s with a fixed forward-lag of 1.5 s, i.e., $K_W = \lfloor 15 f_s/W \rfloor$ and $K_F = \lfloor 1.5 f_s/W \rfloor$. The sample trial for tuning the distribution parameters can be thought of as an initialization step for the estimator prior to its real-time application.

3.2.2.3 Estimation Results

Figure 3.9 shows the results of our estimation framework. Row A contains the estimated TRFs for the encoding model. The major components of the TRFs are retrieved in the estimates while the ℓ_1-norm penalty in Eq. (3.1) has significantly denoised these components as compared with the original noisy versions in row A of Fig. 3.8. Row B in Fig. 3.9 displays the extracted magnitudes of the M100 components from the estimated TRFs at each instance. The attention marker in this case is defined as the magnitude of the M100 component, where the M100 component is calculated as the minimum value of the TRF estimate around the 100 ms lag. Notice that there is a significant statistical difference between the extracted M100 components for the attended and unattended speakers in case 1, while the estimated M100 components are highly variable in case 2 and do not show a strong attention modulation effect.

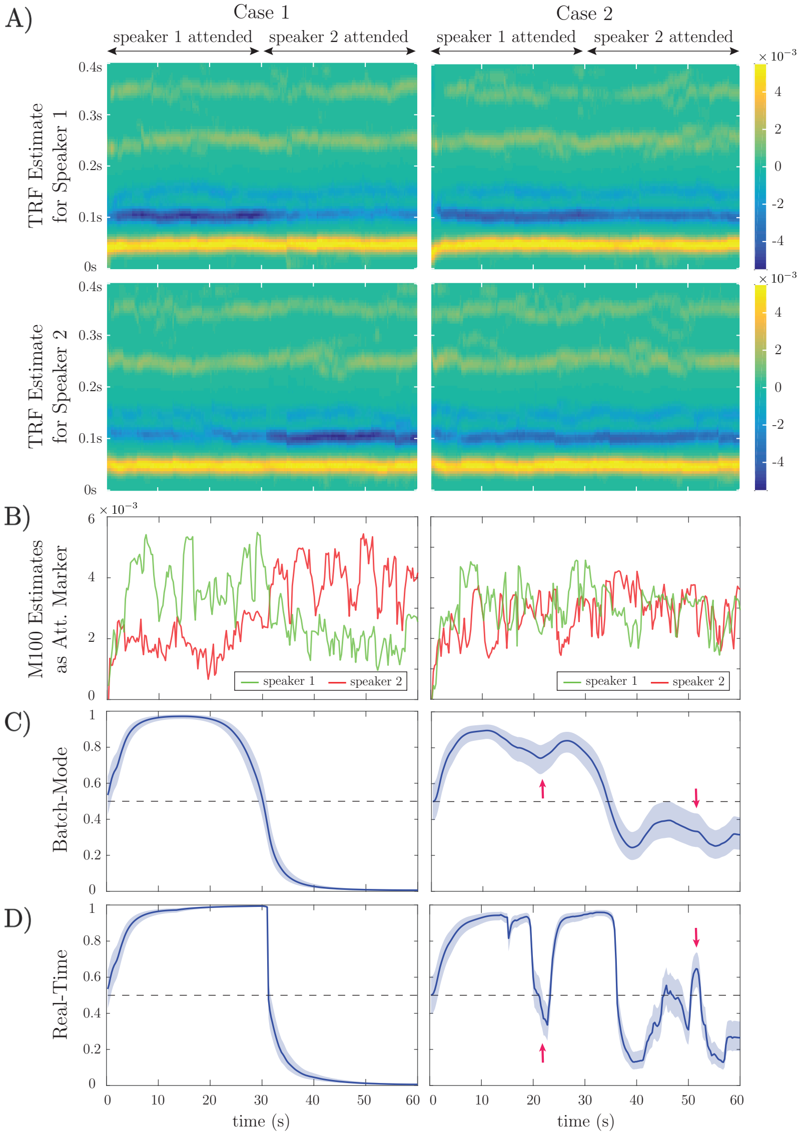

Figure 3.9: Estimation results of application to simulated MEG data: A) Estimated TRFs for case 1 (strong modulation in M100 components) and case 2 (weak modulation in M100 components). B) Estimated M100 magnitudes as the attention markers. C) Outputs of the batch-mode estimator as the estimated probability of attending to speaker 1. D) Outputs of the real-time estimator as the estimated probability of attending to speaker 1. The real-time estimator is less robust to the statistical fluctuations in the extracted M100 components, which can result in misclassifications as shown for two example instances marker by red arrows. However, it follows the general trend of the batch-mode estimator closely despite its online access to data.

Rows C and D of Fig. 3.9 show the output of the batch-mode and real-time state-space estimators, respectively. In case 1, both the batch-mode and real-time estimators perform well in tracking the attentional state. Note that the sharp drop of the attention probability near $\sim 30\,\mathrm{s}$ in Row D is due to the fact that at each instance the real-time estimator does not observe the attention markers beyond the 1.5 s forward lag, whereas the batch-mode estimator estimates the probabilities given the entire trial. In case 2, the batch-mode estimator performs well even though the M100 components are not visually indicative of the attentional state. However, the classification confidence decreases considerably specially in the $(30, 60]$ s interval. The real-time estimator in case 2 closely follows the batch-mode estimator, but is more sensitive to the fluctuations of the extracted M100 components. Thus, its performance undergoes further degradation going from case 1 to 2, as compared with that of the batch-mode estimator. The red arrows in rows C and D of case 2 in Fig. 3.9 mark instances where the less robustness of real-time estimator resulted in misclassifications, while the batch-mode estimator classified the attended speaker correctly.

It is worth noting that as we are using an encoding model in this case, the overall delay in estimating the attentional state is the forward-lag window, i.e., 1.5 s, and unlike the case of using the decoding model, the encoder lag does not contribute to the delay. Our analysis of the effect of K_F on the MSE of the real-time estimator with respect to the batch-mode was nearly identical to that presented for the EEG simulation, and is thus omitted for brevity.

3.2.3 Application to EEG

In this subsection, we apply our real-time attention decoding framework to EEG recordings in a dual-speaker environment. Details of the experimental procedures are given in Section 3.1.6.

3.2.3.1 Preprocessing and Parameter Selection

Both the EEG data and the speech envelopes were downsampled to $f_s = 64\,\text{Hz}$ using an anti-aliasing filter. As the trials had variable lengths, we have considered the first 53 s of each trial for analysis. We have considered consecutive windows of length 0.25 s for decoder estimation, resulting in $W = 16$ samples per window and $K = 212$ instances for each trial. Also, we have considered lags up to 0.25 s for decoder estimation, i.e., $L_d = 16$. The latter is motivated by the results of [45] suggesting that the most relevant decoder components are within the first 0.25 s lags. Prior studies have argued that the effects of auditory attention and speech perception are strongest in the frontal and close-to-ear EEG electrodes [46, 74–76]. We have only considered 28 EEG channels in the decoder estimation problem, i.e., $C = 28$, including the frontal channels Fz, F1-F8, FCz, FC1-FC6, FT7-FT10, C1-C6, and the T complex channels T7 and T8. According to [47], using only this number of electrodes in the decoding process results in nearly the same classification performance as in the case of using all the electrodes. Note that for our real-time setting, a channel selection step can considerably decrease the computational cost and the dimensionality of the decoder estimation step, given that a vector of size

$1+C(L_d+1)$ needs to be updated within each $0.25\,\mathrm{s}$ window.

We have determined the regularization coefficient $\gamma = 0.4$ via cross-validation and the forgetting factor $\lambda = 0.975$, which results in an *effective* data length of 10 s in the estimation of the decoder and is long enough for stable estimation of the decoding coefficients. It is worth noting that small values of λ, and hence small effective data lengths, may result in an under-determined inverse problem, since the dimension of the decoder is given by $1+C(L_d+1)$. Finally, in the FASTA package, we have used a tolerance of 0.01 together with Nesterov's accelerated gradient descent method to ensure that the processing can be done in an online fashion.

In studies involving correlation-based measures, such as [45, 51], the convention is to train attended and unattended decoders/encoders using multiple trials and then use them to calculate the correlation measures over the test trials. The correlation-based attention marker, however, did not produce a statistically significant segregation of the attended and the unattended speakers in our analysis. This discrepancy seems to stem from the fact that the estimated encoders/decoders and the resulting correlations in the aforementioned studies are more informative and robust due to the use of batch-more analysis with multiple trials, as compared to our real-time framework. The ℓ_1-based attention marker, however, resulted in a meaningful statistical separation between the attended and the unattended speakers. Therefore, in what follows, we present our EEG analysis results using the ℓ_1-based attention marker.

The parameters of the state-space models have been set similar to those used in simulations, i.e., $K_W = \lfloor 15 f_s/W \rfloor$, $K_F = \lfloor 1.5 f_s/W \rfloor$, $a_0 = 2.008$, $b_0 = 0.2016$. Considering the 0.25 s lag in the decoder model, the total delay in estimating the attentional state for the real-time system is 1.75 s. For estimating the prior distribution parameters for each subject, we use the first 15s of each trial. As mentioned before, considering the 15 s-long sliding window, we can treat the first 15 s of each trial as a tuning step in which the prior parameters are estimated in a supervised manner and the state-space model parameters are initialized with the values estimated using these initial windows. Thus, similar to the simulations, $\left(\alpha_0^{(a)}, \alpha_0^{(u)}, \beta_0^{(a)}, \beta_0^{(u)}, \mu_0^{(a)}, \mu_0^{(u)}\right)$ for each subject have been set according to the parameters of the two fitted Log-Normal distributions on the ℓ_1-norm of the decoders in the first 15 s of the trials, while choosing large variances for the priors to be non-informative.

3.2.3.2 Estimation Results

Fig. 3.10 shows the results of applying our proposed framework to EEG data. For graphical convenience, the data have been rearranged so that speaker 1 is always attended. The left, middle and right panels correspond to subjects 1, 2, and 3, respectively. For each subject, three example trials have been displayed in rows A, B, and C. Row A includes trials in which the attention marker clearly separates the attended and unattended speakers, while Row C contains trials in which the attention marker fails to do so. Row B displays trials in which on average the ℓ_1-norm of the estimated decoder is larger for the attended speaker; however, occasionally, the attention marker fails to capture the attended speaker.

Consistent with our simulations, the real-time estimates (third graphs in rows

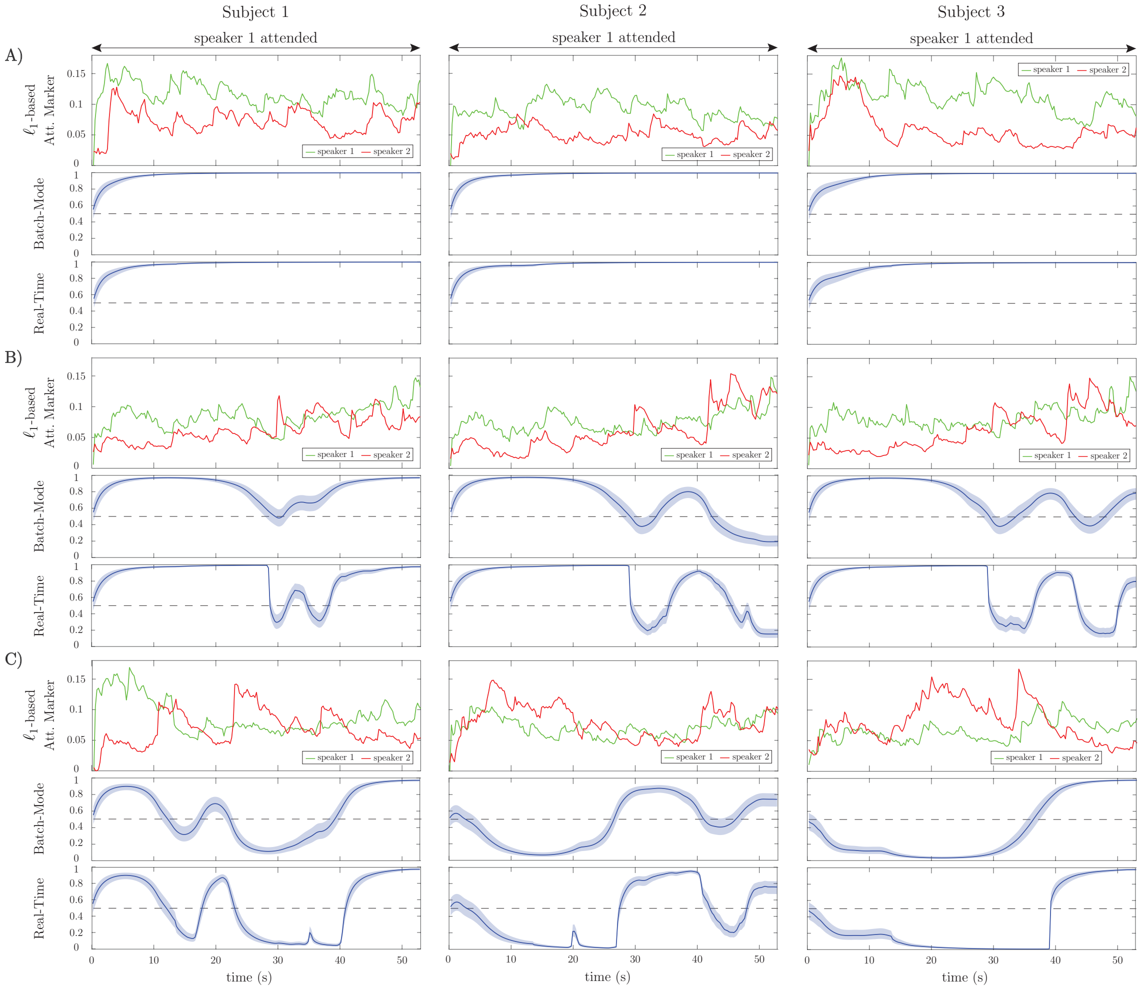

Figure 3.10: Examples of the ℓ_1-based attention markers (first panels), batch-mode (second panels), and real-time (third panels) state-space estimation results for nine selected EEG trials. A) Representative trials in which the attention marker reliably separates the attended and unattended speakers. B) Representative trials in which the attention marker separates the attended and unattended speakers on average over the trial. C) Representative trials in which the attention marker either does not separate the two speakers or results in a larger output for the unattended speaker.

A, B and C) generally follow the output of the batch-mode estimates (second graphs in rows A, B and C). However, the batch-mode estimates yield smoother transitions and larger confidence intervals in general, both of which are due to having access to future observations.

Figure 3.11 shows the effect of forward-lag K_F on the performance of real-time estimates, similar to that shown in Fig. 3.6 for the simulations. The forward-lag K_F is increased from 0 s to 5 s with 0.5 s increments while all the other parameters of the EEG analysis remain the same. The MSE in Fig. 3.11 has been averaged over all trials for each subject. As we observe in the incremental MSE plot, even a 0.5 s lag can significantly decrease the MSE from the case of $K_F = 0$ s (corresponding to the fully real-time setting). Similar to the simulations, we have chosen $K_F = 1.5$ s for the EEG analysis, since the incremental MSE improvements are significant at this lag, and this choice results in a tolerable delay for real-time applications.

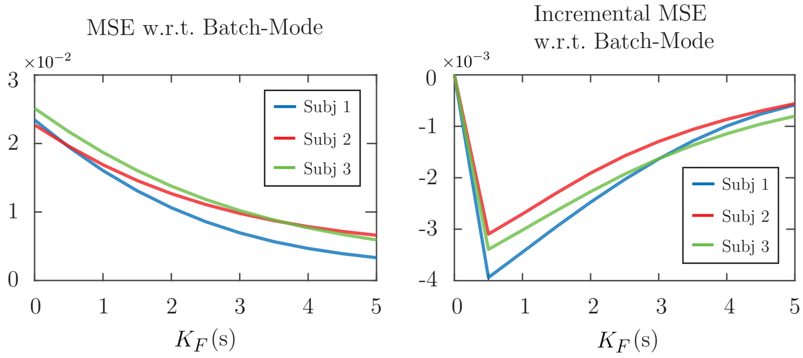

Figure 3.11: Effect of the forward-lag K_F on MSE in application to real EEG data. The left panel shows the MSE with respect to the batch-mode output averaged over all the trials for each subject. The right panel displays the incremental MSE at each lag, from $K_F = 0$ s to $K_F = 5$ s with 0.5 s increments.

Finally, Fig. 3.12 summarizes the *real-time* classification results of our EEG analysis at the group level. Fig. 3.12-A shows a cartoon of the estimated attention probabilities for a generic trial in order to illustrate the classification conventions. We define an instance (i.e., K consecutive windows of length W) to be correctly (incorrectly) classified if the estimated attentional state probability together with

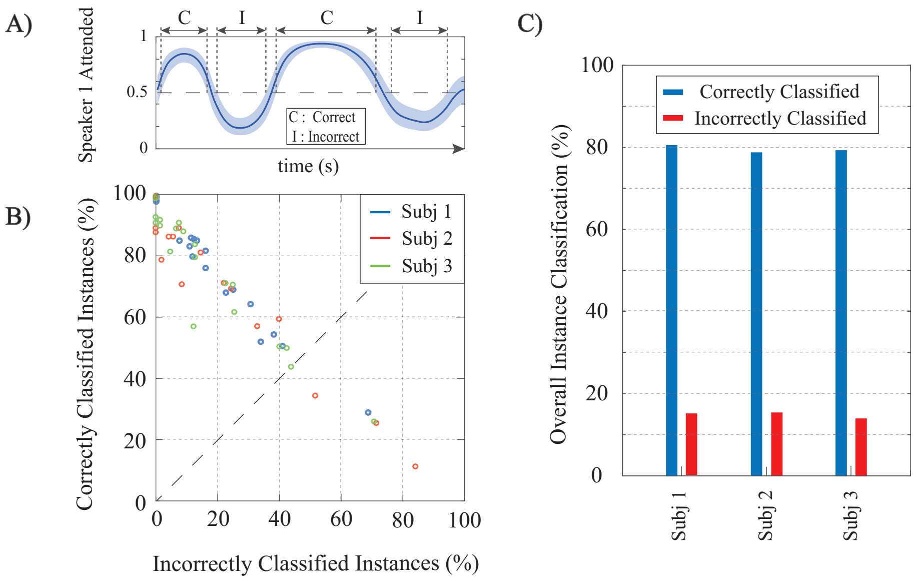

Figure 3.12: Summary of the real-time classification results in application to real EEG data: A) a generic example of the state-space output for a trial illustrating the classification conventions. B) Classification results per trial for all subjects; each circle corresponds to a trial and the subjects are color-coded. The trials falling below the dashed line have more incorrectly classified instances than correctly classified ones. C) Average classification performance over all trials for the three subjects.

its 90% confidence intervals lie above (below) 0.5. If the 90% confidence interval at an instance includes the 0.5 attention probability line, we do not classify it as either correct or incorrect. Figure 3.12-B displays the correctly classified instances (y-axis) versus those incorrectly classified (x-axis) for each trial. The subjects are color-coded and each circle corresponds to one trial. The average classification results over all trials for each subject are shown in Figure 3.12-C. In summary, our framework provides $\sim$ 80% average hit rate and $\sim$ 15% average false-alarm per trial per subject. The group-level hit rate and false alarm rate are respectively given by 79.63% and 14.84%.

3.2.4 Application to MEG

In this subsection, we apply our real-time attention decoding framework to MEG recordings of multiple subjects in a dual-speaker environment. The MEG experimental procedures are discussed in Section 3.1.7.

3.2.4.1 Preprocessing and Parameter Selection

The recorded MEG responses were band-pass filtered between 1 Hz-8 Hz (delta and theta bands), corresponding to the slow temporal modulations in speech [48,49], and downsampled to 200 Hz. MEG recordings, like EEG, include both the stimulus-driven response as well as the background neural activity, which is irrelevant to the stimulus. For the encoding model used in our analysis, we need to extract the stimulus-driven portion of the response, namely the auditory component. In [56,77], a blind source separation algorithm called the Denoising Source Separation (DSS) has been introduced which decomposes the data into temporally uncorrelated components ordered according to their trial-to-trial phase-locking reliability. In doing so, DSS only requires the responses in different trials and not the stimuli. Similar to [51,52], we only use the first DSS component as the auditory component, since it tends to capture a significant amount of stimulus information and to produce a bilateral stereotypical auditory field pattern.

Since DSS is an *offline* algorithm operating on all the data at once, we cannot readily use it for real-time attention decoding. Instead, we apply DSS to the data from pilot trials from each subject in order to calculate the *subject-specific* linear combination of the MEG channels that compose the first DSS component. We then use these channel weights to extract the MEG auditory responses during the constant-attention and attention-switch experiments in a real-time fashion. Note that the MEG sensors are not fixed with respect to the head position across subjects and are densely distributed in space. Therefore, it is not reasonable to use the same MEG channel weights for all subjects. The pilot trials for each subject can thus serve as a training and tuning step prior to the application of our proposed attention decoding framework.

The MEG auditory component extracted using DSS is used as E_t in our encoding model. Similar to our foregoing EEG analysis, we have considered consecutive windows of length 0.25 s resulting in $W = 50$ samples per window and a total number of $K = 240$ instances, at a sampling frequency of 200 Hz. The TRF length, or the total encoder lag, has been set to 0.4 s resulting in $L_e = 80$ in order to include the most significant TRF components [48]. The ℓ_1-regularization parameter γ in Eq. (3.1) has been adjusted to 1 through two-fold cross-validation, and we have chosen a forgetting factor of $\lambda = 0.975$ for capturing the data dynamics resulting in an *effective* data length of 10 s, long enough to ensure estimation stability.

As for the encoder model, we have used a Gaussian dictionary $\mathbf{G_0}$ to enforce smoothness in the TRF estimates. The columns of $\mathbf{G}_0$ consist of overlapping Gaussian kernels with the standard deviation of 20 ms whose means cover the 0 s to 0.4 s lag range with $T_s = 5$ ms increments. The 20 ms standard deviation is consistent with the average full width at half maximum (FWHM) of an auditory MEG evoked response (M50 or M100), empirically obtained from MEG studies [52]. Thus, the overall dictionary discussed in Remark 3.2 takes the form $\mathbf{G} = \text{diag}\,(1, \mathbf{G_0}, \mathbf{G_0})$. Also, similar to [52], we have used the logarithm of the speech envelopes as the regression covariates. Finally, the parameters of the FASTA package in encoder estimation have been chosen similar to those in the foregoing EEG analysis.

The M100 component of the TRF has shown to be more significant for the attended speaker than the unattended speaker [48, 52]. Thus, at each instance k, we extract the magnitude of the negative peak close to the 0.1 s delay in the real-time TRF estimate of each speaker as the attention markers $m_k^{(1)}$ and $m_k^{(2)}$. For the state-space model and the fixed-lag window, we have used the same configuration as in our foregoing EEG analysis, i.e. $K_W = \lfloor 15 f_s/W \rfloor$, $K_F = \lfloor 1.5 f_s/W \rfloor$, $a_0 = 2.008$, and $b_0 = 0.2016$. Note that the total delay in estimating the attentional state is now only 1.5 s, given that we use an encoding model for our MEG analysis. Furthermore, the prior distribution parameters for each subject were chosen according to the two fitted Log-Normal distributions on the extracted M100 values in the first 15 s of the trials, while choosing large variances for the Gamma priors to be non-informative. Similar to the preceding cases, the first 15 s of each trial can be thought of as an initialization stage.

3.2.4.2 Estimation Results

Figure 3.13 shows our estimation results for four sample trials from the constant-attention (cases 1 and 2) and attention-switch (cases 3 and 4) experiments. For graphical convenience, we have rearranged the MEG data such that in the constant-attention experiment, the attention is always on speaker 1, and in the attention-switch experiment, speaker 1 is attended from 0 s to 28 s. Cases 1 and 3 corresponds to trials in which the extracted M100 values for the attended speaker are more significant than those of the unattended speaker during most of the trial duration. Cases 2 and 4, on the other hand, correspond to trials in which the extracted M100 values are not reliable representatives of the attentional state. Row A in Fig. 3.13 shows the estimated TRFs for speakers 1 and 2 in time for each of the four cases. The location of the M100 peaks is shown and tracked with a narrow line (yellow) on the extracted M100 components (blue). The M50 components are also evident as positive peaks occurring around the 50 ms lag. The M50 components do not strongly depend on the attentional state of the listener [48,52,78,79], which is consistent with those shown in Fig. 3.13-A.

Row B in Fig. 3.13 displays the extracted M100 peak magnitudes over time for speakers 1 and 2. The attention modulation effect is more significant in cases 1 and 3. Rows C and D respectively show the batch-mode and real-time estimates of the attentional state based on the extracted M100 values. As expected, the batch-mode output is more robust to the fluctuations in the extracted M100 peak values, with smoother transitions and larger confidence intervals. Despite the poor attention modulation effect in cases 2 and 4, we observe that both the real-time and the batch-mode state-space models show reasonable performance in translating the extracted M100 peak values to a robust measure of the attentional state. This effect is notable in Rows C and D of Case 4. We performed the same analysis as in Fig. 3.11 to assess the effect of the forward-lag parameter K_F. Since the results were

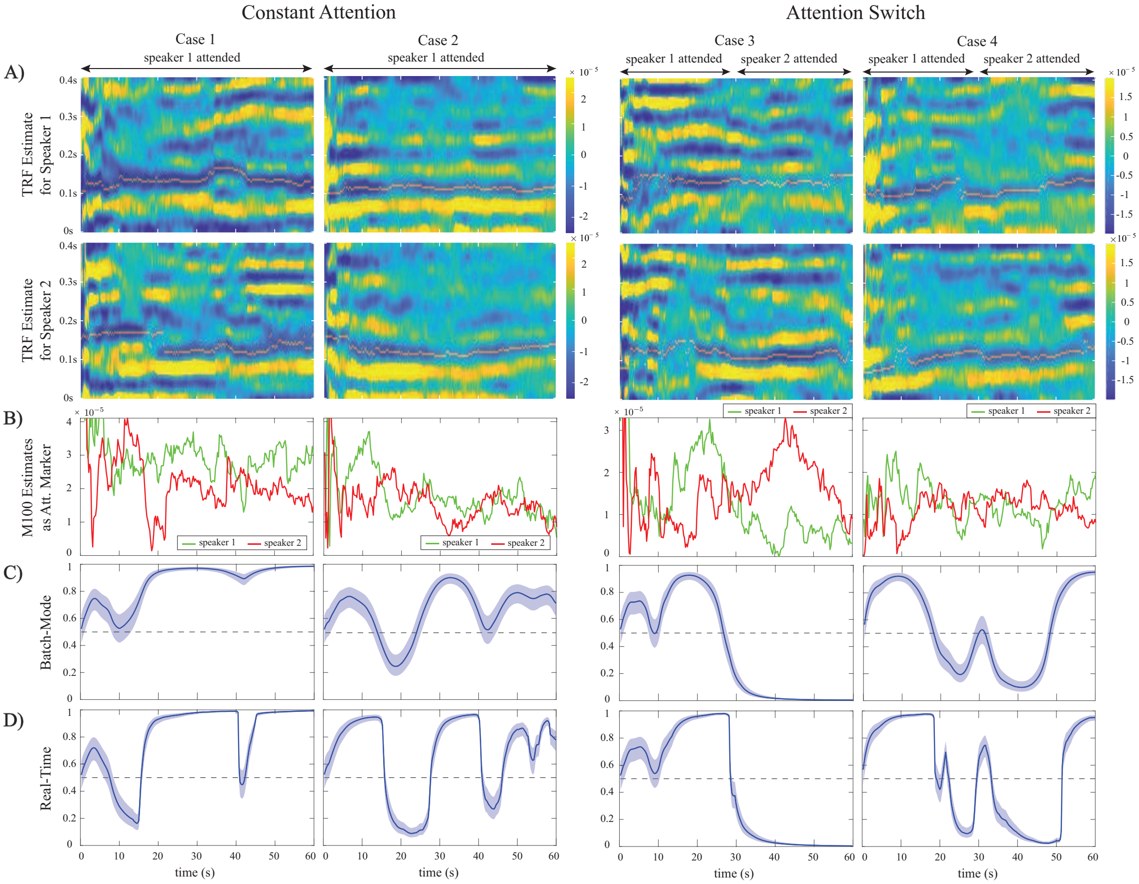

Figure 3.13: Examples from the constant-attention and attention-switch MEG experiments, using the M100 attention marker, for trials with reliable (cases 1 and 3) and unreliable (cases 2 and 4) separation of the attended and unattended speakers: A) TRF estimates for speakers 1 and 2 over time with the extracted M100 peak positions tracked by a narrow yellow line. B) Extracted M100 peak magnitudes over time for speakers 1 and 2 as the attention marker. In cases 1 and 3, the M100 components exhibit a strong modulation effect of the attentional state, i.e., the attended speaker has a larger M100 peak, in contrast to cases 2 and 4, where there is a weak modulation. C) Batch-mode state-space estimates of the attentional state. D) Real-time state-space estimates of the attentional state. The strong or weak modulation effects of attentional state in the extracted M100 components directly affects the classification accuracy and the width of the confidence intervals for both the batch-mode and real-time estimators.

quite similar to those in Figures 3.6 and 3.11, we have omitted them for brevity and chose the same forward-lag of 1.5 s.

Finally, Fig. 3.14 summarizes the *real-time* classification results for the constant-attention (left panels) and attention-switch (right panels) MEG experiments. The classification convention is similar to that used in our EEG analysis, and is illustrated in Fig. 3.14-A for the completeness. For the attention-switch experiment, the 28 s-30 s interval is removed from the classification analysis, as it pertains to a silence period during which the subject is instructed to switch attention. Fig. 3.14-B shows the corresponding classification results, consisting of 36 trials for the constant-attention and 18 trials for the attention-switch experiments. Each circle corresponds to a single trial and the subjects in each experiment are color-coded. The average classification results per trial are shown in Fig. 3.14-C for each subject. The average hit rate and false alarm rates in the constant-attention experiments are respectively given by 71.67% and 20.81%. These quantities for the attention-switch experiment are respectively given by 64.12% and 26.16%, showing a reduction in hit rate and increase in false alarm.

3.3 Discussion

In this work, we have proposed a framework for real-time decoding of the attentional state of a listener in a dual-speaker environment from M/EEG. This framework consists of three modules. In the first module, the encoding/decoding coefficients, relating the neural response to the envelopes of the two speech streams, are estimated in a low-complexity and real-time fashion. Existing approaches for encoder/decoder estimation operate in an offline fashion using multiple experiment

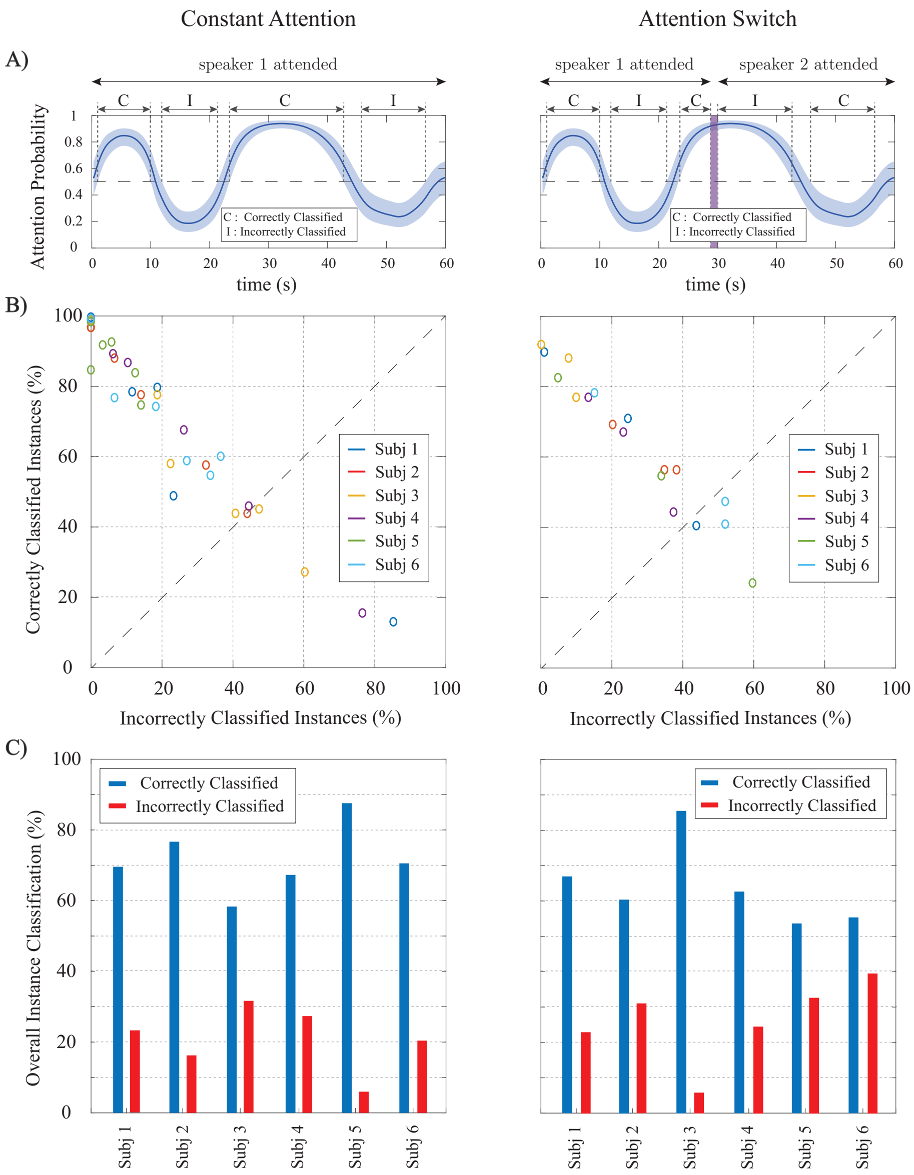

Figure 3.14: Summary of real-time classification results for the constant-attention (left panels) and attention-switch (right panels) MEG experiments: A) a generic instance of the state-space output for a trial illustrating the classification convention. B) Classification results per trial for all subjects; each circle corresponds to a trial and the subjects are color-coded. The trials falling below the dashed line have more incorrectly classified instances than correctly classified ones. C) Average classification performance over all trials for the six subjects.

trials or large training datasets [45, 51, 60, 63], and hence are not suitable for real-time applications. To address this issue, we have integrated the forgetting factor mechanism used in adaptive filtering with ℓ_1-regularization, in order to capture the coefficient dynamics and mitigate overfitting.

In the second module, a function of the estimated encoding/decoding coefficients and the acoustic data, which we refer to as the *attention marker*, is calculated in real-time for each speaker. The role of the attention marker is to provide dynamic features that create statistical separation between the attended and the unattended speakers. Examples of such attention markers include correlation-based measures (e.g. correlation of the acoustic envelopes and their reconstruction from neural response), or measures solely based on the estimated decoding/encoding coefficients (e.g. the ℓ_1-norm of the decoder coefficients or the M100 peak of the encoder).

Finally, the attention marker is passed to the third module consisting of a near real-time state-space estimator. To control the delay in state estimation, we adopt a fixed-lag smoothing paradigm, in which the past and near future data are used to estimate the states. The role of the state-space model is to translate the noisy and highly variable attention markers to robust measures of the attentional state with minimal delay. We have archived a publicly available MATLAB implementation of our framework on the open-source repository GitHub to ease reproducibility [32].

We validated the performance of our proposed framework using simulated EEG and MEG data, in which the ground truth attentional states are known. We also applied our proposed methods to experimentally recorded MEG and EEG data. As for a comparison benchmark, we considered the offline state-space attention decoding approach of [51]. Our MEG analysis showed that although the proposed real-time estimator has access to significantly fewer data points, it closely matches the outcome of the offline state-space estimator in [51], for which the entire data from multiple trials are used for attention decoding. In particular, our analysis of the MEG data in constant-attention conditions revealed a hit rate of $\sim$ 70% and a false alarm rate of $\sim$ 20% at the group level. While the performance is slightly degraded compared to the offline analysis of [51], our algorithms operate in real-time with 1.5s forward delay, over single trials, and using minimal tuning. Similarly, our analysis of EEG data provided $\sim$ 80% hit rate and $\sim$ 15% false alarm rate at a single trial level. These performance measures are slightly degraded compared to the results of offline approaches such as [45].

Our proposed modular design admits the use of any attention-modulated statistic or feature as the attention marker, three of which have been considered in this work. While some attention markers perform better than the rest in certain applications, our goal in this work was to provide different examples of attention markers which can be used in the encoding/decoding models based on the literature, rather than comparing their performance against each other. The choice of the best attention marker that results in the highest classification accuracy is a problem-specific matter. Our modular design allows to evaluate the performance of a variety of attention markers for a given experimental setting, while fixing the encoding/decoding estimation and state-space modules, and to choose one that provides the desired classification performance.

A practical limitation of our proposed methodology in its current form is the need to have access to clean acoustic data in order to form regressors based on the speech envelopes. In a realistic scenario, the speaker envelopes have to be extracted from the noisy mixture of speeches recorded by microphone arrays. Thanks to a number of fairly recent results in attention decoding literature [59–63], it is possible to integrate our methodology with a pre-processing module that extracts the acoustic features of individual speech streams from their noisy mixtures. We view this extension as a future direction of research.

Our proposed framework has several advantages over existing methodologies. First, our algorithms require minimal amount of offline tuning or training. The subject-specific hyperparameters used by the algorithms are tuned prior to real-time application in a supervised manner. The only major offline tuning step in our framework is computing the subject-specific channel weights in the encoding model for MEG analysis in order to extract the auditory component of the neural response. This is due to the fact that the channel locations are not fixed with respect to the head position across subjects. It is worth noting that this step can be avoided if the encoding model treats the MEG channels separately in a multivariate model. Given that recent studies suggest that the M100 component of the encoder obtained from the MEG auditory response is a reliable attention marker [48, 49, 52], we adopted the DSS algorithm for computing the channel weights that compose the auditory response in an offline fashion.

Second, our analysis allows to characterize the performance of the attentional state classification using single trials, which is important for practical applications such as smart hearing aids. Existing studies based on offline algorithms perform classification based on cross-trial performance. For instance, in [45], for each 1 min of test trial, 29 mins of training data are used. In addition, the probabilistic output of our attentional state decoding framework can be used for further statistical analysis and soft-decision mechanisms which are desired in smart hearing aid applications. Finally, the modular design of our framework facilitates its adaptation to more complex auditory scenes (e.g. with multiple speakers and realistic noise and reverberation conditions) and integration of other covariates relevant to real-time applications (e.g. electrooculography measurements).

Chapter 4: Gaussian Mixture Process Noise Inference in State-Space Models with Application to Dynamic Estimation of Temporal Response Functions

State-space modeling is a commonly-used framework for estimation of latent dynamic processes, i.e., the states, under limited observations [80]. The application domains of this approach in time series analysis include control system design [81], tracking [82], finance [83], and most recently neuroscience [16, 84–86]. State-space models (SSMs) often consist of two equations: the state (evolution) equation, to describe the dynamics of the latent process, and the observation equation, to illustrate how the observations are related to the process. These equations are typically described in a parametric fashion using domain-specific expert knowledge of the problem, and parameter estimation is mostly performed via Expectation Maximization (EM) [27, 87] or Variational Inference (VI) [88, 89]. To model the state evolution and measurement uncertainties, additive noise terms are often considered in both the state and observation equations. In most applications, i.i.d. Gaussian statistics are imposed on these terms to account for the aggregate uncertainties and mismatches in the model. Under linear dynamics and observations, Gaussian noise, and fixed model parameters, Minimum Mean Square Error (MMSE) state estimation is conducted by the well-known Kalman filter and smoother [80]. For more general SSMs, Sequential Monte Carlo (SMC) methods can be used for MMSE state estimation [90].

Gaussian statistics is often consistent with empirical histograms of observation noise, which can be estimated from stimulus-free measurements in experimental settings. The noise process driving the state dynamics, often referred to as the process noise, however, does not necessarily follow Gaussian statistics in various real-world applications [91, 92]. This is mainly due to the following two reasons: First, in time series analysis, outliers and abrupt changes in the latent process cannot be properly represented by a Gaussian random variable. Second, the statistics of the process noise are reliant upon how the latent process evolves during the course of the experiment, which heavily depends on the specific experimental requirements, such as the task demand and subject's performance, as well as other exogenous variables not accounted for.

This issue is particularly important in modeling brain function as a latent dynamic process: taking the states to represent the underlying neural circuits that process sensory stimuli, the process noise then consists of both the underlying behaviorally- and stimulus-driven dynamics as well as the background neural activity (not necessarily evoked by the stimulus or behavior), which are typically quite structured and far from being Gaussian. In this context, the state evolution model is more prone to model mismatch and biases, as compared to the observation equation, considering that we generally have more control over the measurement system than the generative mechanism governing the latent process. As a result, the empirical histogram of the process noise (which can be computed from state estimates) could exhibit multimodal morphology, with each mode corresponding to a different exogenous process driving the state dynamics during specific portions of the experiment.

This has led researchers to study SSMs with a Gaussian Mixture (GM) process noise [93–97] considering that a GM can, in principle, approximate any multimodal density [98]. These existing results primarily focus on state estimation and approximation of filtering and smoothing densities under a *fixed* or *known* GM noise density. As such, parameter estimation for a GM process noise in SSMs has not been well-studied. Switching SSMs has been another direction of research in extending linear Gaussian SSMs to cope with nonstationarity, model mismatch, and exogenous processes [89,99–102]. In this approach, several linear Gaussian SSMs are considered to underlie the observed time-series data, which switch place according to a Hidden Markov Model (HMM). Although the filtering and smoothing densities in this model take a GM form, the potential multimodality of the process noise is not explored nor modeled in this approach.

In this work, we fill this gap by developing an EM-based algorithm for estimating the parameters of a GM process noise from the observations in an SSM. The EM algorithm has been widely used for parameter estimation both in state-space modeling [87] and in GM clustering [103], which makes it a promising candidate for our setting. The EM framework in this setting, however, results in intractable expectations for parameter updates. We address this issue by leveraging a SMCEM-type algorithm [104] to approximate the expectations using smoothed particles obtained through SMC. A major drawback of particle smoothing approaches is their excessive computational requirements, or equivalently suffering from sample depletion as the dimension of the target densities grows while fixing the computational costs [105]. As a more scalable alternative, we develop another method of approximating the expectations based on closed-form approximations to the smoothing densities as well as their one-step cross covariances. To this end, we adopt the two-filter formula for smoothing [94] and devise a belief propagation algorithm in our setting. As a result, the computational complexity of the E-step in EM for a GM process noise would be comparable to that of a conventional Gaussian process noise, akin to performing parallel Kalman filtering and smoothing procedures.

To demonstrate the benefits of a GM process noise and the efficacy of the developed estimation framework, we consider the problem of estimating Temporal Response Functions (TRFs) involved in auditory processing [106]. The TRF can be considered as an evolving Finite Impulse Response (FIR) filter which gets convolved with speech features in time, e.g., the speech envelope, to produce the auditory neural response observed through neuroimaging modalities such as electroencephalography (EEG) and magnetoencephalography (MEG) [107]. The TRF framework has resulted in new insights into the mechanisms of speech processing in the brain, specially under competing-speaker environments, i.e., the cocktail party scenario [40, 46, 48, 51]. For instance, TRF components at specific lags may exhibit peaks which *arise*, *persist*, and *disappear* over time according to the attentional state of the listener [108]. The different local dynamics of TRF components under each of these conditions motivates a GM density to capture such evolution patterns.

Dynamic estimation of TRFs was first discussed in [107] using a Recursive Least Square (RLS) algorithm. However, smoothing estimates and state-space modeling are more robust than RLS and filtering estimates in performing a comprehensive dynamic analysis of TRFs when data from multiple trials is available. Thus, we study dynamic estimation of TRFs using SSMs and apply our SSM framework with a GM process noise to both simulated and experimentally recorded MEG data under a dual-speaker environment where the subject switches attention between the two speakers at will. The results show that our proposed algorithm can effectively recover the multimodal structure of the process noise from SSM observations, and that having a richer and more realistic representation of the process noise allows to capture the TRF dynamics more precisely and more consistent with the subjects' behavioral reports, as compared to the conventional Gaussian SSM or RLS estimation. While our proposed framework is motivated by and applied to data from auditory experiments, it is applicable to general state-space modeling problems in which states exhibit heterogeneous and recurring local dynamic patterns.

The rest of the paper is organized as follows: Section 4.1 presents the SSM formulation with a GM process noise and defines the main parameter estimation problem. The corresponding EM algorithm and the two approximation methodologies are discussed in Section 4.2, followed by our simulation and real data analysis results in Section 4.3. Finally, Section 4.4 includes our concluding remarks.

4.1 Problem Formulation

Consider the following generic discrete-time SSM with additive noise:

$$\begin{cases} \mathbf{x}_n = f_n(\mathbf{x}_{n-1}) + \mathbf{w}_n \\ \\ \mathbf{y}_n = g_n(\mathbf{x}_n) + \mathbf{v}_n \end{cases} \tag{4.1}$$

where $\mathbf{x}_n \in \mathbb{R}^{d_x}$ and $\mathbf{y}_n \in \mathbb{R}^{d_y}$ represent the states and the observations at time index n, respectively. We assume that the functional forms of $f_n(.)$ and $g_n(.)$ are known and fixed for $n = 1, \dots, N$ using domain-specific knowledge of the problem. Following our introductory discussion, let $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ for the observation noise. Also, to represent the process noise, consider a GM with M mixture components and parameter set $\Theta := \{p_{1:M}, \boldsymbol{\mu}_{1:M}, \boldsymbol{\Sigma}_{1:M}\}$ containing the mixture probabilities $p_{1:M}$, mean vectors $\boldsymbol{\mu}_{1:M}$, and covariance matrices $\boldsymbol{\Sigma}_{1:M}$. We model the state dynamics over $K := N/W$ consecutive non-overlapping windows of length W. Within each window $i \in \{1, \dots, K\}$, the process noise is drawn from one of the mixture components, which we denote by $z_i \in \{1, \dots, M\}$. Therefore, we have $\mathbf{w}_n \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$ for $n = (i–1)W+1, \dots, iW$, and we consider the z_i's to be i.i.d. with $\mathrm{P}(z_i = m) = p_m$ for $m = 1, \dots, M$. In other words, z_i determines the active mixture component that governs the state dynamics in window i. This can also be interpreted as a jumping or switching Gaussian process noise. Note that for special case of $W = 1$, the resulting model could in principle approximate any arbitrary i.i.d. process noise $\mathbf{w}_n$ as it is fitting a GM model to the process noise. In this case, the labels z_i of mixture components can vary at the same rate as that of the states and observations.

Let $\boldsymbol{\mathcal{Y}}_{n_1}^{n_2}$ denote the set of observations from n_1 to n_2, i.e., $\mathbf{y}_{n_1:n_2}$, and similarly define $\boldsymbol{\mathcal{X}}_{n_1}^{n_2}$ and $\mathcal{Z}_{i_1}^{i_2}$ for $\mathbf{x}_{n_1:n_2}$ and $z_{i_1:i_2}$, respectively. Our goal is to estimate the GM process noise parameters Θ from SSM observations $\boldsymbol{\mathcal{Y}}_1^N$. As estimation of observation noise covariance $\mathbf{R}$ in EM is straightforward [87], we assume $\mathbf{R}$ to be fixed for convenience and will briefly review the update equations for $\mathbf{R}$ in Section 4.2, if it need to be estimated from the observed data. As mentioned in the introduction, $\mathbf{R}$ can also be estimated from stimulus-free conditions. Finally, we adopt the Maximum Likelihood (ML) estimation framework to estimate Θ as follows:

$$\widehat{\Theta}_{\mathsf{ML}} := \arg\max_{\Theta} \mathrm{P}\left(\boldsymbol{\mathcal{Y}}_1^N \,\middle|\, \Theta\right) \tag{4.2}$$

Despite its simple statement, the problem of Eq. (4.2) is challenging due to the difficulties in computing the optimization argument, i.e., data likelihood, in a computationally scalable fashion. We will address this challenge in the forthcoming section.

4.2 Parameter Estimation

We use the EM algorithm as a solution method for the ML problem in (4.2). The EM framework provides an iterative procedure to update the estimated parameter set with the guarantee that at iteration $(\ell+1)$ we have

$$\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_1^N \,\middle|\, \widehat{\Theta}^{(\ell+1)}\right) \geq \mathrm{P}\left(\boldsymbol{\mathcal{Y}}_1^N \,\middle|\, \widehat{\Theta}^{(\ell)}\right) \tag{4.3}$$

where $\widehat{\Theta}^{(\ell)}$ is the parameter set estimate from the ℓ^{th} iteration [27]. The EM algorithm guarantees convergence to a local maximum, and most of the work on escaping the undesirable local maxima in EM theory have focused on providing an informed initialization of the algorithm [109, 110]. As will be explained in Section 4.3, we will use fixed-interval smoothed estimates based on a Gaussian model to choose $\widehat{\Theta}^{(0)}$ and initialize the algorithm.

Let $\mathcal{H} = \left\{\mathcal{Z}_1^K, \boldsymbol{\mathcal{X}}_1^N\right\}$ denote the set of latent variables in the SSM, which includes the states and the labels of active mixture component in each window. The EM algorithm performs the following two steps at the $(\ell+1)^{\text{th}}$ iteration and repeats them until convergence to a parameter estimate $\widehat{\Theta}$:

$$\begin{cases} \text{E-step}: Q\left(\Theta \middle| \widehat{\Theta}^{(\ell)}\right) = \mathrm{E}_{\mathcal{H}}\left\{\log \mathrm{P}\left(\boldsymbol{\mathcal{Y}}_1^N, \mathcal{H} \middle| \Theta\right) \middle| \boldsymbol{\mathcal{Y}}_1^N, \widehat{\Theta}^{(\ell)}\right\} \\ \text{M-step}: \widehat{\Theta}^{(\ell+1)} = \arg\max_{\Theta} Q\left(\Theta \middle| \widehat{\Theta}^{(\ell)}\right) \end{cases} \tag{4.4}$$

where the surrogate function $Q\big(\Theta \big| \widehat{\Theta}^{(\ell)}\big)$ is a lower bound on the data log-likelihood. The expectation in E-step is over the conditional density $\mathcal{H} \,|\, \boldsymbol{\mathcal{Y}}_1^N, \widehat{\Theta}^{(\ell)}$. As all of the following expectations are also conditioned on $\boldsymbol{\mathcal{Y}}_1^N$ and $\widehat{\Theta}^{(\ell)}$, we drop the conditioning in the notation for convenience, but keep the expectation subscript to denote the random variable with respect to which the expectation is taken. Also, hereafter the subscript (i,j) represents the time index of the j^{th} sample in the i^{th} window, i.e., $n = (i-1)W + j$ for brevity. The two steps of the EM algorithm in Eq. (4.4) in our setting can be expressed as follows:

E-Step: The surrogate function in the SSM of Section 4.1 is computed as

$$
\begin{aligned}
Q\Big(\Theta\Big|\widehat{\Theta}^{(\ell)}\Big) &= \mathrm{E}_{\mathcal{H}}\Big\{\log \mathrm{P}\Big(\boldsymbol{\mathcal{Y}}_1^N, \mathcal{H} \,\Big|\, \Theta\Big)\Big\} \qquad (4.5)\\
&= \mathrm{E}_{\mathcal{H}}\Big\{\log \mathrm{P}\Big(\mathcal{Z}_1^K\Big|\Theta\Big) + \log \mathrm{P}\Big(\boldsymbol{\mathcal{X}}_1^N\Big|\mathcal{Z}_1^K, \Theta\Big)\Big\} + \mathsf{cst}.\\
&= \sum_{i=1}^{K}\sum_{m=1}^{M}\mathrm{E}_{\mathcal{H}}\Bigg\{\mathbb{1}_{\{z_i=m\}}\Bigg(\log p_m + \sum_{j=1}^{W}\log \pi_{(i,j),m}\Bigg)\Bigg\} + \mathsf{cst}.,
\end{aligned}
$$

where $\mathbb{1}_{\{.\}}$ denotes the indicator function, $\mathsf{cst}.$ stands for all the terms not dependent on Θ and may vary from one equation to another, and $\pi_{(i,j),m}$ is defined as

$$
\pi_{(i,j),m} := \mathrm{P}\left(\mathbf{x}_{(i,j)} \,\middle|\, \mathbf{x}_{(i,j\text{-}1)}, z_i = m, \Theta\right), \qquad (4.6)
$$

which is computed based on the Gaussian density for $\mathbf{w}_{(i,j)}$ in Eq. (4.1) when $z_i = m$. If we decompose the conditional expectation in Eq. (4.5) into two iterated conditional expectations with respect to $\boldsymbol{\mathcal{X}}_1^N \,|\, \boldsymbol{\mathcal{Y}}_1^N, \widehat{\Theta}^{(\ell)}$ and $\mathcal{Z}_1^K \,|\, \boldsymbol{\mathcal{X}}_1^N, \widehat{\Theta}^{(\ell)}$ (where $\boldsymbol{\mathcal{Y}}_1^N$ is dropped in the latter due to conditional independence), this equation can be written as

$$
Q\Big(\Theta\Big|\widehat{\Theta}^{(\ell)}\Big) = \sum_{i=1}^{K}\sum_{m=1}^{M}\mathrm{E}_{\mathbf{x}}\Bigg\{\widehat{\epsilon}_{i,m}^{(\ell)}\Bigg(\log p_m + \sum_{j=1}^{W}\log \pi_{(i,j),m}\Bigg)\Bigg\} + \mathsf{cst}., \qquad (4.7)
$$

where $\widehat{\epsilon}_{i,m}^{(\ell)}$ is the membership probability and can be expressed using the Bayes' rule as:

$$
\begin{aligned}
\widehat{\epsilon}_{i,m}^{(\ell)} := \mathrm{P}\Big(z_i = m\Big|\boldsymbol{\mathcal{X}}_1^N, \widehat{\Theta}^{(\ell)}\Big) &= \mathrm{P}\Big(z_i = m\Big|\boldsymbol{\mathcal{X}}_{(i,0)}^{(i,W)}, \widehat{\Theta}^{(\ell)}\Big) \qquad (4.8)\\
&= \frac{\hat{p}_m^{(\ell)} \prod_{j=1}^{W} \widehat{\pi}_{(i,j),m}^{(\ell)}}{\sum_{m'=1}^{M} \hat{p}_{m'}^{(\ell)} \prod_{j=1}^{W} \widehat{\pi}_{(i,j),m'}^{(\ell)}},
\end{aligned}
$$

The variable $\widehat{\pi}_{(i,j),m}^{(\ell)}$ is defined similarly to (4.6) but for $\Theta = \widehat{\Theta}^{(\ell)}$, which makes $\widehat{\epsilon}_{i,m}^{(\ell)}$ a constant with respect to Θ in Eq. (4.7).

M-Step: In this step, we maximize the log-likelihood lower bound with respect to Θ. Differentiating (4.7) with respect to Θ, enforcing the condition $\sum_{m=1}^{M} p_m = 1$, and invoking the dominated convergence theorem to change the order of expectation and differentiation, we obtain the following parameter updates for $m = 1, \ldots, M$:

$$\hat{p}_m^{(\ell+1)} = \frac{1}{K}\sum_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\left\{\widehat{\epsilon}_{i,m}^{(\ell)}\right\}, \tag{4.9}$$

$$\widehat{\boldsymbol{\mu}}_m^{(\ell+1)} = \frac{\sum_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\Big\{\widehat{\epsilon}_{i,m}^{(\ell)} \sum_{j=1}^{W} \mathbf{v}_{(i,j)}\Big\}}{W\sum_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\{\widehat{\epsilon}_{i,m}^{(\ell)}\}}, \tag{4.10}$$

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_m^{(\ell+1)} &= \frac{\sum\limits_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\left\{\widehat{\epsilon}_{i,m}^{(\ell)}\sum\limits_{j=1}^{W}\Big(\mathbf{v}_{(i,j)}-\widehat{\boldsymbol{\mu}}_m^{(\ell+1)}\Big)\Big(\mathbf{v}_{(i,j)}-\widehat{\boldsymbol{\mu}}_m^{(\ell+1)}\Big)^{\top}\right\}}{W\sum_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\{\widehat{\epsilon}_{i,m}^{(\ell)}\}} \\ &= \frac{\sum\limits_{i=1}^{K}\mathrm{E}_{\mathbf{x}}\left\{\widehat{\epsilon}_{i,m}^{(\ell)}\sum\limits_{j=1}^{W}\mathbf{v}_{(i,j)}\mathbf{v}_{(i,j)}^{\top}\right\}}{W\sum_{i=1}^{K} \mathrm{E}_{\mathbf{x}}\{\widehat{\epsilon}_{i,m}^{(\ell)}\}} - \widehat{\boldsymbol{\mu}}_m^{(\ell+1)}\Big(\widehat{\boldsymbol{\mu}}_m^{(\ell+1)}\Big)^{\top},\end{aligned} \tag{4.11}$$

where $\mathbf{v}_{(i,j)} = \mathbf{x}_{(i,j)} - f_{(i,j)}\big(\mathbf{x}_{(i,j\text{-}1)}\big)$.

Remark 4.1. If the covariance matrix $\mathbf{R}$ of the Gaussian observation noise in (4.1) also needs to be estimated from $\boldsymbol{\mathcal{Y}}_1^N$, it can be included in the parameter set Θ. The update formula for $\widehat{\mathbf{R}}^{(\ell+1)}$ in the EM framework then becomes [111]

$$\widehat{\mathbf{R}}^{(\ell+1)} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{E}_{\mathbf{x}}\Big\{\left(\mathbf{y}_n - g_n(\mathbf{x_n})\right)\left(\mathbf{y}_n - g_n(\mathbf{x_n})\right)^{\top}\Big\} \tag{4.12}$$

In the definition of $\widehat{\epsilon}_{i,m}^{(\ell)}$ in Eq. (4.8), both the numerator and the denominator include exponential functions of the states. Therefore, the conditional expectations in Eq. (4.7) and in the update equations above are intractable even if the joint smoothing density $\boldsymbol{\mathcal{X}}_1^N \mid \boldsymbol{\mathcal{Y}}_1^N, \widehat{\Theta}^{(\ell)}$ is known in closed-form [97]. The rest of this section is dedicated to developing two approaches to approximately compute these expectations. Readers who are primarily interested in the algorithmic developments may proceed with the rest of this section, whereas those who find the application of the proposed GM inference methods to simulated and experimentally-recorded data immediately more useful may skip to Section 4.3.

4.2.1 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 $\mathbf{x}^{(u)}_{(i,0):(i,W)}$ for $u = 1, \ldots, U$ denote a number of U sample paths, i.e., particles, with corresponding weights of $\omega^{(u)}_i$ inside the i^{th} window to approximate the joint smoothing density $\boldsymbol{\mathcal{X}}^{(i,W)}_{(i,0)} \,|\, \boldsymbol{\mathcal{Y}}^N_1, \widehat{\Theta}^{(\ell)}$ for $i\!=\!1, \ldots, K$. Using this particle approximation, the update equations of the M-step become:

$$\hat{p}^{(\ell+1)}_m \approx \frac{1}{K} \sum_{i=1}^{K} \sum_{u=1}^{U} \omega^{(u)}_i \widehat{\epsilon}^{(\ell,u)}_{i,m}, \tag{4.13}$$

$$\widehat{\boldsymbol{\mu}}^{(\ell+1)}_m \approx \frac{\sum_{i=1}^K \sum_{u=1}^U \omega^{(u)}_i \widehat{\epsilon}^{(\ell,u)}_{i,m} \sum_{j=1}^W \mathbf{v}^{(u)}_{(i,j)}}{W \sum_{i=1}^K \sum_{u=1}^U \omega^{(u)}_i \widehat{\epsilon}^{(\ell,u)}_{i,m}}, \tag{4.14}$$

$$\widehat{\boldsymbol{\Sigma}}^{(\ell+1)}_m \approx \frac{\displaystyle\sum_{i=1}^{K} \sum_{u=1}^{U} \omega^{(u)}_i \widehat{\epsilon}^{(\ell,u)}_{i,m} \sum_{j=1}^{W} \left(\mathbf{v}^{(u)}_{(i,j)} - \widehat{\boldsymbol{\mu}}^{(\ell+1)}_m\right)\left(\mathbf{v}^{(u)}_{(i,j)} - \widehat{\boldsymbol{\mu}}^{(\ell+1)}_m\right)^{\top}}{W \sum_{i=1}^K \sum_{u=1}^U \omega^{(u)}_i \widehat{\epsilon}^{(\ell,u)}_{i,m}}, \tag{4.15}$$

where $\mathbf{v}^{(u)}_{(i,j)} = \mathbf{x}^{(u)}_{(i,j)} - f_{(i,j)}\left(\mathbf{x}^{(u)}_{(i,j\text{-}1)}\right)$, and $\widehat{\epsilon}^{(\ell,u)}_{i,m}$ is defined similarly to $\widehat{\epsilon}^{(\ell)}_{i,m}$ in (4.8) with $\widehat{\pi}^{(\ell)}_{(i,j),m}$'s evaluated at $\boldsymbol{\mathcal{X}}^{(i,W)}_{(i,0)} = \mathbf{x}^{(u)}_{(i,0):(i,W)}$ and $\Theta = \widehat{\Theta}^{(\ell)}$ in Eq. (4.6). Particle smoothing approaches are SMC methods which provide the sample paths $\mathbf{x}^{(u)}_{(i,0):(i,W)}$ and their respective weights $\omega^{(u)}_i$ [90]. The class of algorithms using SMC within EM for SSMs are referred to as SMCEM [104]. A forward-backward particle smoothing algorithm is presented in Alg. 7 as an example of how the approximating particles can be computed.

Remark 4.2. In general, particle smoothing approaches are computationally intensive, especially for high dimensional problems, which limits their application compared to particle filtering methods. In our setting, densities of dimension $d_x(W+1)$ have to be approximated by particles. The forward-backward method in Alg. 7 simply re-weights the filtering particles according to future observations and incurs an $\mathcal{O}(U^2)$ cost. The two-filter particle smoother [112] samples the particles in the smoothing step but has a similar computational cost. In [112], an approximation based on spatial-index methods is introduced to reduce the computational cost to $\mathcal{O}(U \log U)$. Finally, a particle smoothing method with $\mathcal{O}(U)$ cost (similar to that of particle filtering) is developed in [113]. However, it operates under the assumption of minimal posterior dependence between $\mathbf{x}_{n-1}$ and $\mathbf{x}_{n+1}$ when sampling for the smoothing density of $\mathbf{x}_n$.

4.2.2 Approach 2: Closed-Form Approximations

In this section, we consider a linear SSM, i.e., $f_n(\mathbf{x}_{n-1}) = \mathbf{A}_n\mathbf{x}_{n-1}$ and $g_n(\mathbf{x}_n) = \mathbf{C}_n\mathbf{x}_n$ in (4.1), to exploit the GM formulation of the smoothing densities [97]. Techniques such as the extended Kalman filter [80] or the unscented Kalman filter [114] are often used to approximate the general state-space model of Eq. (4.1) with a linear model. We introduce an approximation to the expectations in the M-step which

Algorithm 7 A Forward-Backward Particle Smoothing Alg.

```
Inputs: state-space model in (4.1) and parameter estimate $\widehat{\Theta}^{(\ell)}$.
Output: sample paths $\mathbf{x}_{(i,0):(i,W)}^{(u)}$ and their weights $\omega_i^{(u)}$.
Initialize $\mathbf{x}_0^{(u)}$ and their filtering weights $\bar{\omega}_0^{(u)} = 1/U$.
for $i = 1 : K$ do
    Sample $z_i^{(u)}$ according to $\hat{p}_{1:M}^{(\ell)}$.
    Sample $\mathbf{x}_{(i,0):(i,W)}^{(u)}$ using $\mathbf{x}_{(i,0)}^{(u)}$ as the starting point and $z_i^{(u)}$ as the active Gaussian component.
    $\widetilde{\omega}_i^{(u)} = \prod_{j=1}^{W} \mathrm{P}\left(\mathbf{y}_{(i,j)}^{(u)} \,\middle|\, \mathbf{x}_{(i,j)}^{(u)}\right)$.
    Normalize the weights such that $\sum_{u=1}^{U} \bar{\omega}_i^{(u)} = 1$.
    Resample $\mathbf{x}_{(i,W)}^{(u)}$ for next window according to $\bar{\omega}_i^{(u)}$.
end for
Initialize the smoothing weights $\omega_K^{(u)} = \widetilde{\omega}_K^{(u)}$.
for $i = K - 1 : 1$ do
    $\omega_i^{(u)} = \widetilde{\omega}_i^{(u)} \sum_{u'=1}^{U} \frac{\mathrm{P}\left(\mathbf{x}_{(i+1,1)}^{(u')} \,\middle|\, \mathbf{x}_{(i,W)}^{(u)}, \widehat{\Theta}^{(\ell)}\right) \omega_{i+1}^{(u')}}{\sum_{u''=1}^{U} \mathrm{P}\left(\mathbf{x}_{(i+1,1)}^{(u')} \,\middle|\, \mathbf{x}_{(i,W)}^{(u'')}, \widehat{\Theta}^{(\ell)}\right) \widetilde{\omega}_i^{(u'')}}$.
end for
```

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 [87]. 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 $\widehat{\epsilon}_{i,m}^{(\ell)}$ in the update formulas of (4.9)-(4.11) around the mean of the smoothing densities. In other words, $\widehat{\epsilon}_{i,m}^{(\ell)} \approx \bar{\epsilon}_{i,m}^{(\ell)}$ where $\bar{\epsilon}_{i,m}^{(\ell)}$ is computed similarly to (4.8) with $\widehat{\pi}_{(i,j),m}^{(\ell)}$'s evaluated at $\boldsymbol{\mathcal{X}}_1^N = \bar{\mathbf{x}}_{1:N} := \mathrm{E}_{\mathbf{x}}\{\mathbf{x}_{1:N}\}$ and $\Theta = \widehat{\Theta}^{(\ell)}$ in Eq. (4.6).

Remark 4.3. 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 $\widehat{\epsilon}^{(\ell)}_{i,m}$ 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 $\widehat{\epsilon}^{(\ell)}_{i,m}$ 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 our numerical experiments in Section 4.3, the 0th order approximation suffices for our applications of interest.

It is known that for a linear SSM with Gaussian mixture noise, the filtering and smoothing densities also take Gaussian mixture forms [94, 97]. Let

$$\mathrm{P}\left(\boldsymbol{\mathcal{X}}^{n}_{n\text{-}1}\middle|\boldsymbol{\mathcal{Y}}^{N}_{1}, \widehat{\Theta}^{(\ell)}\right) = \sum_{\gamma=1}^{\Gamma_{\mathsf{S}}} \rho_n^{(\mathsf{s},\gamma)} \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_{n\text{-}1} \\ \\ \mathbf{x}_n \end{bmatrix}; \boldsymbol{\mu}_n^{(\mathsf{s},\gamma)}, \boldsymbol{\Sigma}_n^{(\mathsf{s},\gamma)}\right) \tag{4.16}$$

be the one-step joint smoothing density at time n, where the superscript s identifies smoothing parameters, and Γ_{S} is the number of mixture components forming the smoothing density. Taking $\bar{\epsilon}^{(\ell)}_{i,m}$ out of the expectations, the M-step update equations become:

$$\hat{p}_m^{(\ell+1)} \approx \frac{1}{K} \sum_{i=1}^{K} \bar{\epsilon}^{(\ell)}_{i,m}, \tag{4.17}$$

$$\widehat{\boldsymbol{\mu}}_m^{(\ell+1)} \approx \frac{\sum\limits_{i=1}^{K} \bar{\epsilon}_{i,m}^{(\ell)} \sum\limits_{j=1}^{W} \widetilde{\mathbf{A}}_{(i,j)} \sum\limits_{\gamma=1}^{\Gamma_{\mathsf{s}}} \rho_{(i,j)}^{(\mathsf{s},\gamma)} \boldsymbol{\mu}_{(i,j)}^{(\mathsf{s},\gamma)}}{W \sum_{i=1}^{K} \bar{\epsilon}_{i,m}^{(\ell)}}, \tag{4.18}$$

$$\widehat{\boldsymbol{\Sigma}}_m^{(\ell+1)} \approx \frac{\sum\limits_{i=1}^{K} \bar{\epsilon}_{i,m}^{(\ell)} \sum\limits_{j=1}^{W} \widetilde{\mathbf{A}}_{(i,j)} \sum\limits_{\gamma=1}^{\Gamma_{\mathsf{s}}} \rho_{(i,j)}^{(\mathsf{s},\gamma)} \left(\boldsymbol{\Sigma}_{(i,j)}^{(\mathsf{s},\gamma)} + \boldsymbol{\mu}_{(i,j)}^{(s,\gamma)} \left(\boldsymbol{\mu}_{(i,j)}^{(s,\gamma)} \right)^{\top} \right) \widetilde{\mathbf{A}}_{(i,j)}^{\top}}{W \sum_{i=1}^{K} \bar{\epsilon}_{i,m}^{(\ell)}} - \widehat{\boldsymbol{\mu}}_m^{(\ell+1)} \left(\widehat{\boldsymbol{\mu}}_m^{(\ell+1)} \right)^{\top}, \tag{4.19}$$

where $\widetilde{\mathbf{A}}_{(i,j)} = [\text{-}\mathbf{A}_{(i,j)}, \mathbf{I}_{d_x}]$ with $\mathbf{I}_{d_x}$ denoting the identity matrix of dimension d_x. Another approach to approximately compute the expectations in the update equations (4.9)-(4.11) is to use the Laplace approximation [115]. This approach, however, requires the computation of the GM joint smoothing density $\boldsymbol{\mathcal{X}}_{(i,0)}^{(i,W)} \,|\, \boldsymbol{\mathcal{Y}}_1^N, \widehat{\Theta}^{(\ell)}$ and would be 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. (4.16), i.e., $\left\{ \rho_n^{(\mathsf{s},\gamma)}, \boldsymbol{\mu}_n^{(\mathsf{s},\gamma)}, \boldsymbol{\Sigma}_n^{(\mathsf{s},\gamma)} \right\}$, have to be estimated for $n = 1, \ldots, N$ in the E-step. In Section II.D of [97], 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 [94], 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 [94] is that either $\mathbf{C}_n$ 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 (4.16) in closed-form. Since all of the following densities are conditioned on $\widehat{\Theta}^{(\ell)}$ similar to Eq. (4.16), we hereafter drop the conditioning in our notation for convenience.

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

$$\mathrm{P}\left(\mathbf{x}_{(i,0)}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{(i,0)}\right)=\sum_{\gamma=1}^{\Gamma_{\mathsf{F}}}\rho_{(i,0)}^{(\mathsf{f},\gamma)}\mathcal{N}\left(\mathbf{x}_{(i,0)};\boldsymbol{\mu}_{(i,0)}^{(\mathsf{f},\gamma)},\boldsymbol{\Sigma}_{(i,0)}^{(\mathsf{f},\gamma)}\right) \tag{4.20}$$

where superscript f identifies forward filtering parameters and Γ_{F} is the number of mixtures forming the filtering density at the end of each window. Also, let the unnormalized backward information filter [94] at the end of the i^{th} window be defined as

$$\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_{iW}^{N}\,\middle|\,\mathbf{x}_{iW}\right)\propto\sum_{\gamma=1}^{\Gamma_{\mathsf{B}}}\beta_{iW}^{(\gamma)}\exp\left\{-\frac{1}{2}\mathbf{x}_{iW}^{\top}\mathbf{B}_{iW}^{(\gamma)}\mathbf{x}_{iW}+\mathbf{x}_{iW}^{\top}\mathbf{b}_{iW}^{(\gamma)}\right\} \tag{4.21}$$

where Γ_{B} is the number of exponential components forming the information filter at the end of each window. Note that Eq. (4.21) is not a density in $\mathbf{x}$. Considering the independence of z_i and $\boldsymbol{\mathcal{Y}}_1^{(i,0)}$, the two-filter formula for window i in our switching GM process noise model can be written as

$$\begin{aligned}\mathrm{P}\left(\boldsymbol{\mathcal{X}}_{(i,j\text{-}1)}^{(i,j)}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{N}\right)&=\frac{\mathrm{P}\left(\boldsymbol{\mathcal{X}}_{(i,j\text{-}1)}^{(i,j)},\boldsymbol{\mathcal{Y}}_{(i,1)}^{N}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{(i,0)}\right)}{\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_{(i,1)}^{N}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{(i,0)}\right)}\\&=\frac{1}{\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_{(i,1)}^{N}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{(i,0)}\right)}\sum_{m=1}^{M}\hat{p}_m^{(\ell)}\times\\&\mathrm{P}\left(\mathbf{x}_{(i,j\text{-}1)},\boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,j\text{-}1)}\,\middle|\,\boldsymbol{\mathcal{Y}}_1^{(i,0)},\mathfrak{z}_i^m\right)\mathrm{P}\left(\mathbf{x}_{(i,j)}\,\middle|\,\mathbf{x}_{(i,j\text{-}1)},\mathfrak{z}_i^m\right)\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_{(i,j)}^{N}\,\middle|\,\mathbf{x}_{(i,j)},\mathfrak{z}_i^m\right)\end{aligned} \tag{4.22}$$

where $\mathfrak{z}_i^m$ stands for the event $\{z_i=m\}$. The leftmost term in the last line of Eq. (4.22) is the forward filter and represents an unnormalized filtering density, which we express as:

$$\mathrm{P}\left(\mathbf{x}_{(i,j)}, \boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,j)} \middle| \boldsymbol{\mathcal{Y}}_{1}^{(i,0)}, \mathfrak{z}_i^m\right) = \sum_{\gamma=1}^{\Gamma_{\mathsf{F}}} \rho_{(i,j),m}^{(\mathsf{f},\gamma)} \mathcal{N}\left(\mathbf{x}_{(i,j)}; \boldsymbol{\mu}_{(i,j),m}^{(\mathsf{f},\gamma)}, \boldsymbol{\Sigma}_{(i,j),m}^{(\mathsf{f},\gamma)}\right) \tag{4.23}$$

for $j = 1, \ldots, W$ and compute it through the following unnormalized forward recursion in j:

$$\begin{aligned}\mathrm{P}\left(\mathbf{x}_{(i,j)}, \boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,j)} \middle| \boldsymbol{\mathcal{Y}}_{1}^{(i,0)}, \mathfrak{z}_i^m\right) = \int & \mathrm{P}\left(\mathbf{y}_{(i,j)} \middle| \mathbf{x}_{(i,j)}\right) \mathrm{P}\left(\mathbf{x}_{(i,j)} \middle| \mathbf{x}_{(i,j\text{-}1)}, \mathfrak{z}_i^m\right) \\ & \mathrm{P}\left(\mathbf{x}_{(i,j\text{-}1)}, \boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,j\text{-}1)} \middle| \boldsymbol{\mathcal{Y}}_{1}^{(i,0)}, \mathfrak{z}_i^m\right) d\mathbf{x}_{(i,j\text{-}1)}\end{aligned} \tag{4.24}$$

The recursion is initialized by the filtering density in Eq. (4.20) at window i. This results in the following forward filter parameter updates:

$$\begin{cases} \widetilde{\boldsymbol{\mu}} = \mathbf{A}_{(i,j)} \boldsymbol{\mu}_{(i,j\text{-}1),m}^{(\mathsf{f},\gamma)} + \widehat{\boldsymbol{\mu}}_m^{(\ell)} \\ \widetilde{\boldsymbol{\Sigma}} = \mathbf{A}_{(i,j)} \boldsymbol{\Sigma}_{(i,j\text{-}1),m}^{(\mathsf{f},\gamma)} \mathbf{A}_{(i,j)}^{\top} + \widehat{\boldsymbol{\Sigma}}_m^{(\ell)} \\ \mathbf{H} = \widetilde{\boldsymbol{\Sigma}} \mathbf{C}_{(i,j)}^{\top} \left(\mathbf{C}_{(i,j)} \widetilde{\boldsymbol{\Sigma}} \mathbf{C}_{(i,j)}^{\top} + \mathbf{R}\right)^{-1} \\ \boldsymbol{\mu}_{(i,j),m}^{(\mathsf{f},\gamma)} = \widetilde{\boldsymbol{\mu}} + \mathbf{H}\left(\mathbf{y}_{(i,j)} - \mathbf{C}_{(i,j)} \widetilde{\boldsymbol{\mu}}\right) \\ \boldsymbol{\Sigma}_{(i,j),m}^{(\mathsf{f},\gamma)} = \left(\mathbf{I} - \mathbf{H}\mathbf{C}_{(i,j)}\right) \widetilde{\boldsymbol{\Sigma}} \\ \rho_{(i,j),m}^{(\mathsf{f},\gamma)} = \rho_{(i,j\text{-}1),m}^{(\mathsf{f},\gamma)} \mathcal{N}\left(\mathbf{y}_{(i,j)}; \mathbf{C}_{(i,j)} \widetilde{\boldsymbol{\mu}}, \mathbf{C}_{(i,j)} \widetilde{\boldsymbol{\Sigma}} \mathbf{C}_{(i,j)}^{\top} + \mathbf{R}\right) \end{cases} \tag{4.25}$$

and filtering density at time (i, j) is computed from Eq. (4.23) as

$$\mathrm{P}\left(\mathbf{x}_{(i,j)} \middle| \boldsymbol{\mathcal{Y}}_{1}^{(i,j)}\right) \propto \sum_{m=1}^{M} \hat{p}_m \mathrm{P}\left(\mathbf{x}_{(i,j)}, \boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,j)} \middle| \boldsymbol{\mathcal{Y}}_{1}^{(i,0)}, \mathfrak{z}_i^m\right) \tag{4.26}$$

Next, we represent the unnormalized backward information filter, i.e., the rightmost term in the last line of Eq. (4.22), as

$$\mathrm{P}\left(\boldsymbol{\mathcal{Y}}_{(i,j)}^{N} \middle| \mathbf{x}_{(i,j)}, \mathfrak{z}_i^m\right) \propto \sum_{\gamma=1}^{\Gamma_{\mathsf{B}}} \beta_{(i,j),m}^{(\gamma)} \exp\left\{-\frac{1}{2} \mathbf{x}_{(i,j)}^{\top} \mathbf{B}_{(i,j),m}^{(\gamma)} \mathbf{x}_{(i,j)} + \mathbf{x}_{(i,j)}^{\top} \mathbf{b}_{(i,j),m}^{(\gamma)}\right\} \tag{4.27}$$

where we enforce the normalization $\sum_{\gamma=1}^{\Gamma_{\mathsf{B}}}\sum_{m=1}^{M}\beta^{(\gamma)}_{(i,j),m}=1$. Note that this normalization is applied to avoid numerical instabilities while performing the recursions and does not change the final smoothing density of Eq. (4.22), which has to be eventually normalized. The backward filter in Eq. (4.27) can be computed through the following recursion [94]:

$$\begin{aligned}\mathrm{P}\Big(\boldsymbol{\mathcal{Y}}^{N}_{(i,j)}\Big|\mathbf{x}_{(i,j)},\mathfrak{z}_i^m\Big)=\int\mathrm{P}\big(\mathbf{y}_{(i,j)}\big|\mathbf{x}_{(i,j)}\big)\mathrm{P}\big(\mathbf{x}_{(i,j+1)}\big|\mathbf{x}_{(i,j)},\mathfrak{z}_i^m\big)&\\ \mathrm{P}\Big(\boldsymbol{\mathcal{Y}}^{N}_{(i,j+1)}\Big|\mathbf{x}_{(i,j+1)},\mathfrak{z}_i^m\Big)d\mathbf{x}_{(i,j+1)}&\end{aligned}\tag{4.28}$$

and is initialized by the density of Eq. (4.21) at window i. This results in the following parameter updates for the backward filter:

$$\begin{cases}\overline{\boldsymbol{\Sigma}}=\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\left(\mathbf{I}+\mathbf{B}^{(\gamma)}_{(i,j+1),m}\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\right)\\ \overline{\boldsymbol{\mu}}=\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\mathbf{b}^{(\gamma)}_{(i,j+1),m}+\widehat{\boldsymbol{\mu}}^{(\ell)}_m\\ \mathbf{B}^{(\gamma)}_{(i,j),m}=\mathbf{C}^{\top}_{(i,j)}\mathbf{R}^{-1}\mathbf{C}_{(i,j)}\\ \qquad\qquad+\mathbf{A}^{\top}_{(i,j+1)}\left[\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\right)^{-1}-\overline{\boldsymbol{\Sigma}}^{-1}\right]\mathbf{A}_{(i,j+1)}\\ \mathbf{b}^{(\gamma)}_{(i,j),m}=\mathbf{C}^{\top}_{(i,j)}\mathbf{R}^{-1}\mathbf{y}_{(i,j)}-\mathbf{A}^{\top}_{(i,j+1)}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\right)^{-1}\widehat{\boldsymbol{\mu}}^{(\ell)}_m\\ \qquad\qquad+\mathbf{A}^{\top}_{(i,j+1)}\overline{\boldsymbol{\Sigma}}^{-1}\overline{\boldsymbol{\mu}}\\ \beta^{(\gamma)}_{(i,j),m}\propto\beta^{(\gamma)}_{(i,j+1),m}\sqrt{\frac{|\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m|}{|\overline{\boldsymbol{\Sigma}}|}}\exp\left\{-\frac{1}{2}\left(\widehat{\boldsymbol{\mu}}^{(\ell)}_m\right)^{\top}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_m\right)^{-1}\widehat{\boldsymbol{\mu}}^{(\ell)}_m\right\}\\ \qquad\qquad\times\exp\left\{\frac{1}{2}\overline{\boldsymbol{\mu}}^{\top}\overline{\boldsymbol{\Sigma}}^{-1}\overline{\boldsymbol{\mu}}\right\}\end{cases}\tag{4.29}$$

and the overall backward filter in the beginning of window i can be computed from Eq. (4.27) as

$$\mathrm{P}\Big(\boldsymbol{\mathcal{Y}}^{N}_{(i,0)}\Big|\mathbf{x}_{(i,0)}\Big)=\sum_{m=1}^{M}\hat{p}^{(\ell)}_{M}\,\mathrm{P}\Big(\boldsymbol{\mathcal{Y}}^{N}_{(i,0)}\Big|\mathbf{x}_{(i,0)},\mathfrak{z}_i^m\Big)\tag{4.30}$$

Using Eqs. (4.23) and (4.27), the parameters of the joint GM smoothing density in Eq. (4.22) are computed as:

$$\begin{cases} \gamma'' = (\gamma - 1)M\Gamma_{\mathsf{B}} + (m-1)\Gamma_{\mathsf{B}} + \gamma' \\ \mathbf{S}_{11} = \left(\boldsymbol{\Sigma}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\right)^{\text{-}1} + \mathbf{A}^{\top}_{(i,j)}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1}\mathbf{A}_{(i,j)} \\ \mathbf{S}_{12} = \mathbf{S}^{\top}_{21} = -\mathbf{A}^{\top}_{(i,j)}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1} \\ \mathbf{S}_{22} = \mathbf{B}^{(m)}_{(i,j),\gamma'} + \left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1} \\ \boldsymbol{\Sigma}^{(\mathsf{s})}_{(i,j),\gamma''} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}^{\text{-}1} \\ \mathbf{u}_1 = \left(\boldsymbol{\Sigma}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\right)^{\text{-}1}\boldsymbol{\mu}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma} - \mathbf{A}^{\top}_{(i,j)}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1}\widehat{\boldsymbol{\mu}}^{(\ell)}_{m} \\ \mathbf{u}_2 = \left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1}\widehat{\boldsymbol{\mu}}^{(\ell)}_{m} + \mathbf{b}^{(m)}_{(i,j),\gamma'} \\ \boldsymbol{\mu}^{(\mathsf{s})}_{(i,j),\gamma''} = \boldsymbol{\Sigma}^{(\mathsf{s})}_{(i,j),\gamma''}\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \\ \rho^{(\mathsf{s})}_{(i,j),\gamma''} \propto \rho^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\,\hat{p}^{(\ell)}_{m}\,\beta^{(m)}_{(i,j),\gamma'}\sqrt{\frac{|\boldsymbol{\Sigma}^{(\mathsf{s})}_{(i,j),\gamma''}|}{|\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}||\boldsymbol{\Sigma}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}|}} \\ \qquad \times \exp\left\{\text{-}\frac{1}{2}\left(\boldsymbol{\mu}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\right)^{\top}\left(\boldsymbol{\Sigma}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\right)^{\text{-}1}\boldsymbol{\mu}^{(\mathsf{f},m)}_{(i,j\text{-}1),\gamma}\right\} \\ \qquad \times \exp\left\{\text{-}\frac{1}{2}\left(\widehat{\boldsymbol{\mu}}^{(\ell)}_{m}\right)^{\top}\left(\widehat{\boldsymbol{\Sigma}}^{(\ell)}_{m}\right)^{\text{-}1}\widehat{\boldsymbol{\mu}}^{(\ell)}_{m}\right\} \\ \qquad \times \exp\left\{\frac{1}{2}\left(\boldsymbol{\mu}^{(\mathsf{s})}_{(i,j),\gamma''}\right)^{\top}\left(\boldsymbol{\Sigma}^{(\mathsf{s})}_{(i,j),\gamma''}\right)^{\text{-}1}\boldsymbol{\mu}^{(\mathsf{s})}_{(i,j),\gamma''}\right\} \end{cases} \tag{4.31}$$

where we have $\gamma \in \{1, \ldots, \Gamma_{\mathsf{F}}\}$, $m \in \{1, \ldots, M\}$, and $\gamma' \in \{1, \ldots, \Gamma_{\mathsf{B}}\}$. This brings the total number of mixture components in the joint smoothing density of Eq. (4.22) to $\Gamma_{\mathsf{F}} \times M \times \Gamma_{\mathsf{B}}$. As number of mixture components grows exponentially in SSMs with GM noise components [93], limiting them is a crucial step for practical purposes. To this end, in forming the density of Eq. (4.16), the number of mixture components obtained from Eq. (4.22) are reduced to Γ_{S} prior to updating the parameters in the M-step. In this work, we choose Γ_{S} components from the density of Eq. (4.22) with the largest mixture probabilities for simplicity. However, more accurate mixture reduction algorithms are available and developed in [93, 116, 117], but with additional computational costs. It is worth noting that calculations corresponding to the weights $\rho^{(\mathsf{f})}$'s in Eq. (4.25), β's in (4.29), and $\rho^{(\mathsf{s})}$'s in (4.31) should be performed in log-scale to avoid numerical errors in practice.

Algorithm 8 summarizes the steps for calculating the smoothing density parameters in Eq. (4.16) for $n = 1, \ldots, N$. Note that in the case of an unknown observation covariance matrix $\mathbf{R}$, the smoothing densities in Eq. (4.16) can be replaced in the expression of Eq. (4.12) to provide a closed-form update for $\widehat{\mathbf{R}}^{(\ell+1)}$.

4.2.3 Model Selection

An important issue in applications of GMs for clustering is the choice of the number of mixtures M. A variety of model selection criteria have been used in the literature of Gaussian mixtures including the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Independent Component Analysis (ICA) [118–121], 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 $\widetilde{\omega}_i^{(u)}$'s in Algorithm 7 as

$$\log \mathrm{P}\big(\boldsymbol{\mathcal{Y}}_1^N\big) \approx \textstyle\sum_{i=1}^K \log\Big(\sum_{u=1}^U \widetilde{\omega}_i^{(u)}\Big) \tag{4.32}$$

In Approach 2, using the unnormalized filtering densities in closed-form approximation, the log-likelihood in our model can be computed based on [122] as

$$\begin{aligned} \log \mathrm{P}\big(\boldsymbol{\mathcal{Y}}_1^N\big) &= \textstyle\sum_{i=1}^K \log\Big(\sum_{m=1}^M \widehat{p}_m^{(\ell)} \mathrm{P}\Big(\boldsymbol{\mathcal{Y}}_{(i,1)}^{(i,W)} \Big| \boldsymbol{\mathcal{Y}}_1^{(i,0)}, \mathfrak{z}_i^m\Big)\Big) \\ &\approx \textstyle\sum_{i=1}^K \log\Big(\sum_{m=1}^M \widehat{p}_m^{(\ell)} \Big(\sum_{\gamma=1}^{\Gamma_{\mathsf{F}}} \rho_{(i,W),m}^{(\mathsf{f},\gamma)}\Big)\Big) \end{aligned} \tag{4.33}$$

where the last line is derived from integrating the unnormalized filtering density in Eq. (4.23).

4.3 Results

In this section, we demonstrate the utility of our proposed algorithms in estimating TRFs from auditory neural responses to speech, using both simulated and experimentally-recorded MEG data. Before doing so, we will give an overview of the TRF model and how its estimation can be posed within our GM SSM framework.

4.3.1 The TRF Model

Consider a cocktail party setting [40], in which a subject is listening to two speakers simultaneously, but only attending to one of the speakers. While the subject is performing this task, his/her neural response is recorded using MEG. Let $y_t \in \mathbb{R}$ denote the auditory component of the neural response at time $t \in \{1, \ldots, T\}$, extracted from multichannel MEG recordings [56, 123]. Also, let $s_t^{(q)}$ be a speech

Algorithm 8 Two-Filter Gaussian Mixture Smoothing Alg.

```
Inputs: linear state-space model in (4.1), parameter estimate $\widehat{\Theta}^{(\ell)}$, and component
    limits $\Gamma_{\mathsf{F}}$, $\Gamma_{\mathsf{B}}$, and $\Gamma_{\mathsf{S}}$.
Output: smoothing density parameters $\rho^{(\mathsf{s})}_{n,\gamma}$, $\boldsymbol{\mu}^{(\mathsf{s})}_{n,\gamma}$, $\boldsymbol{\Sigma}^{(\mathsf{s})}_{n,\gamma}$ in (4.16) for $n \in \{1, \ldots, N\}$
    and $\gamma \in \{1, \ldots, \Gamma_{\mathsf{S}}\}$.
Initialize the filtering density in (4.20) at $n=0$ as the prior on $\mathbf{x}_0$.
for $i = 1 : K$ do
    Run forward recursions of (4.25) for $m = 1, \ldots, M$ in window $i$ starting from
    (4.20) and store the parameters.
    Compute the filtering density at $n = iW$ from (4.26).
    Out of $\Gamma_{\mathsf{F}} \times M$ mixture components in the filtering density, keep the $\Gamma_{\mathsf{F}}$ ones
    with largest probabilities as initialization for window $i+1$.
end for
Initialize the backward filter as $\mathrm{P}\big(\mathbf{y}_N \big| \mathbf{x}_N\big)$, i.e., $\beta_{N,1} = 1$, $\mathbf{B}_{N,1} = \mathbf{C}_N^{\top}\mathbf{R}^{-1}\mathbf{C}_N$,
and $\mathbf{b}_{N,1} = \mathbf{C}_N^{\top}\mathbf{R}^{-1}\mathbf{y}_N$.
for $i = K : 1$ do
    Run backward recursions of (4.29) for $m = 1, \ldots, M$ in window $i$ starting
    from (4.21).
    Run smoothing algorithm of (4.31) in window $i$ using backward filtering parameters
    and the stored forward filtering parameters.
    Out of $\Gamma_{\mathsf{F}} \times M \times \Gamma_{\mathsf{B}}$ smoothing mixture components, store the $\Gamma_{\mathsf{S}}$ ones with
    the largest probabilities for smoothing densities of (4.16) in window $i$.
    Compute the overall backward filter at $n=(i-1)W$ from (4.30).
    Out of $\Gamma_{\mathsf{B}} \times M$ backward filtering components, keep the $\Gamma_{\mathsf{B}}$ ones corresponding
    to the most significant mixture components of $\mathrm{P}\Big(\boldsymbol{\mathcal{X}}_{(i,0)}^{(i,1)} \Big| \boldsymbol{\mathcal{Y}}_1^N\Big)$ as initialization for
    window $i-1$.
end for
Output the computed smoothing parameters of (4.16).
```

feature of speaker $q \in \{1, 2\}$ at time t, e.g., the acoustic envelope, and denote by $\mathbf{s}_t^{(q)} = [s_t^{(q)}, \ldots, s_{t-L-1}^{(q)}]^{\top} \in \mathbb{R}^L$ the vector containing the previous L features up to (and including) time t. In this work, we consider $s_t^{(q)}$ to be the acoustic envelope in log scale, which is known to be a reliable predictor of the neural response [107]. Other features such as phoneme representations, word frequency measures, and semantic composition have also been considered in the literature [124–126], and can also be included in $s_t^{(q)}$. A widely-used linear stimulus-response model is given by:

$$y_t = \mathbf{s}_t^\top \widetilde{\boldsymbol{\tau}}_t + v_t \tag{4.34}$$

where $\widetilde{\boldsymbol{\tau}}_t = \left[\widetilde{\boldsymbol{\tau}}_t^{(1)}; \widetilde{\boldsymbol{\tau}}_t^{(2)}\right] \in \mathbb{R}^{2L}$ is the concatenation of $\widetilde{\boldsymbol{\tau}}_t^{(1)}$ and $\widetilde{\boldsymbol{\tau}}_t^{(2)}$ as the TRFs at time t corresponding to speakers 1 and 2, respectively. Also, $\mathbf{s}_t = \left[\mathbf{s}_t^{(1)}; \mathbf{s}_t^{(2)}\right] \in \mathbb{R}^{2L}$ is the concatenation of the speech feature vectors at time t, and v_t represents the observations noise. In light of this model, and as mentioned in the introduction, the TRF $\widetilde{\boldsymbol{\tau}}_t^{(q)}$ can be thought of as the impulse response of a linear, but time-varying, system representing the neural activity and taking as input the speech features of speaker q, for $q = 1, 2$. Existing results in auditory neuroscience [48, 79, 86, 106–108] have focused on studying the behavioral significance of the various peaks in the TRF. For instance, the TRF exhibits an early positive peak at around 50 ms, referred to as the M50 component, which is known to represent the encoding of the acoustic envelope. A later negative peak at around 100 ms lag, referred to as the M100 component, has shown to have an attentional modulation effect, so that it appears to have a higher magnitude for the attended speaker's TRF, compared to the unattended speaker's TRF. The M50 component is attributed to the effect of early auditory processing in the brain and is equally represented in both speakers' TRFs, while the M100 component represents the later processing stages segregate the attended speaker from the unattended one [48]. Dynamic estimation of the TRFs can thus provide insights into the underlying neural dynamics that process speech in the cocktail party setting, and has significant implications for the design of non-invasive brain-machine interface devices involving auditory processing, such as the emerging 'smart' hearing aid technology that utilizes neural signals to steer the hearing aid parameters in real-time.

We assume $v_t \sim \mathcal{N}(0, \sigma^2)$ and define the nominal observation SNR as $10\log_{10}\left(\bar{E}/\sigma^2\right)$, where $\bar{E}$ is the average of the signal component in Eq. (4.34) over the trial of length T. It is common to consider a piecewise-constant approximation to the TRFs over consecutive non-overlapping time windows of length t_0, which is comparable to the length of the TRF L. In other words, $\widetilde{\boldsymbol{\tau}}_t = \boldsymbol{\tau}_n$ for $t \in \{(n-1)t_0+1, \ldots, nt_0\}$ and $n \in \{1, \ldots, N\}$ where $N = T/t_0$ is assumed to be an integer without loss of generality. We then define $\mathbf{y}_n = [y_{(n\text{-}1)t_0\text{+}1}, \ldots, y_{nt_0}]^\top$, $\mathbf{S}_n = [\mathbf{s}_{(n\text{-}1)t_0\text{+}1}, \ldots, \mathbf{s}_{nt_0}]$, and $\mathbf{v}_n = [v_{(n\text{-}1)t_0\text{+}1}, \ldots, v_{nt_0}]^\top$. In [107], dynamic estimation of TRFs was first discussed using a regularized RLS framework. First, the TRFs are represented over a dictionary $\mathbf{G}$, i.e, $\boldsymbol{\tau}_n^{(q)} = \mathbf{G}\mathbf{x}_n^{(q)}$, in order to enforce smoothness in the lag domain [46,48]. The dynamic TRF estimation framework of [107] can be stated as:

$$\begin{cases} \widehat{\mathbf{x}}_n = \arg\min_{\mathbf{x}\in\mathbb{R}^{2L}} \sum_{i=1}^{n} \lambda^{n-i} \left\| \mathbf{y}_i - \mathbf{S}_i^\top \widetilde{\mathbf{G}}\mathbf{x} \right\|_2^2 + \gamma h(\mathbf{x}) \\ \widehat{\boldsymbol{\tau}}_n = \widetilde{\mathbf{G}}\widehat{\mathbf{x}}_n \end{cases} \tag{4.35}$$

where $\lambda \in (0,1)$ is the forgetting factor, γ is the regularization coefficient, $h(.)$ can either be an ℓ_1 or ℓ_2 penalty [127], and $\widetilde{\mathbf{G}} = \mathrm{diag}(\mathbf{G}, \mathbf{G})$ is a block diagonal matrix with $\mathbf{G}$ containing the dictionary atoms. Similar to [86,107], we consider a Gaussian dictionary $\mathbf{G} \in \mathbb{R}^{L\times D}$ where the D columns of $\mathbf{G}$ are shifted Gaussian kernels. The parameter λ in Eq. (4.35) induces a trade-off between adaptivity and robustness of TRF estimation.

The estimate in (4.35), however, is a filtering estimate by design and is suited for real-time estimation of TRFs. For a more precise dynamic analysis of the TRFs in an off-line fashion, SSMs have the advantage of providing smoothed estimates and directly modeling the evolution of the TRFs through the state equation. We use the SSM below to represent the TRF dynamics and its relation to the neural response:

$$\begin{cases} \mathbf{x}_n = \alpha \mathbf{x}_{n-1} + \mathbf{w}_n \\ \boldsymbol{\tau}_n = \widetilde{\mathbf{G}} \mathbf{x}_n \\ \mathbf{y}_n = \mathbf{S}_n^\top \boldsymbol{\tau}_n + \mathbf{v}_n \end{cases} \tag{4.36}$$

where $\alpha \in (0, 1)$ controls the nominal rate of change of the TRF, similar to the effect of the forgetting factor λ in Eq. (4.35) for the RLS framework. In [128, 129], a correspondence between α and λ has been discussed which can result in the same filtering estimates of the SMM in Eq. (4.36) with Gaussian noise and the RLS model in Eq. (4.35), without any penalization. The parameter α can either be estimated in the EM framework as in [87], or it can be set based on the domain-specific knowledge of the problem to provide a desired adaptivity-robustness trade-off. The estimated TRFs in (4.36) are computed from the smoothing estimates as $\widehat{\boldsymbol{\tau}}_n = \widetilde{\mathbf{G}} \widehat{\mathbf{x}}_{n|N}$.

In the following two subsections we demonstrate the advantages of the proposed GM SSM inference in TRF estimation from both simulated and experimentally-recorded MEG data, by assuming a GM density for $\mathbf{w}_n$ in Eq. (4.36). We consider the RLS framework of Eq. (4.35) and the smoothed estimates from a linear Gaussian model (Eq. (4.36) with Gaussian $\mathbf{w}_n$) as benchmarks.

4.3.2 Application to Simulated MEG

Consider a 90 s long cocktail party experiment, in which the subject is listening to two speakers simultaneously and is instructed to switch attention between the two every 15 s starting at time 7.5 s. We synthesize the putative TRF dynamics as shown in Fig. 4.3.2-A, based on relevance of the different TRF peaks discussed in Section 4.3.1. We use a sampling rate of $F_s = 100$ Hz and a length of 0.25 s for the TRFs, i.e., $L = 0.25F_s$. Let $\mathbf{G}$ be a dictionary consisting of five Gaussian atoms with variances of 0.018 whose means are separated by 50 ms increments starting from a lag of 0 ms to 200 ms. This results in $\mathbf{G} \in \mathbb{R}^{25\times 5}$ and $\mathbf{x}_n \in \mathbb{R}^{10}$ in Eqs. (4.35) and (4.36). Furthermore, consider a piecewise-constant model for the TRFs over windows of length 300 ms resulting in $N = 300$ TRF samples over the trial for each speaker. Fig. 4.3.2-A shows the synthesized TRF heatmaps for speakers 1 and 2, where the corresponding states $\mathbf{x}_n$'s are designed such that the M50 component stays relatively constant for the two speakers, the M100 component is modulated by the attentional state, and a common high-latency component at 200 ms varies independently of the subject's attention. Fig. 4.3.2-B shows two snapshots of the TRF of speaker 2 at 10 s, when speaker 2 is attended, and at 85 s, when speaker 1 is attended. It is worth noting that the corresponding states in Fig. 4.3.2-A are not generated from an SSM such as the one in (4.36). However, the relatively smooth temporal changes of the TRFs in Fig. 4.3.2-A (representing neural activity in controlled experimental conditions) makes the SSM of Eq. (4.36) a suitable candidate for dynamic TRF analysis. Indeed, the TRF components at lags of 100 ms and 200ms exhibit heterogeneous dynamics across the trial, including periods of increasing, decreasing, and remaining relatively constant, which model the changes in auditory state throughout the experiment. As mentioned in the introduction, such dynamics can be modeled using a multimodal process noise density as in Eq. (4.36). Fig. 4.3.2-C shows the histogram of true $\mathbf{w}_n$ samples in (4.36) along the 3rd state dimension of speaker 2's TRF (corresponding to the M100 component). The process noise samples are computed as $\widehat{\mathbf{w}}_n = \mathbf{x}_n\text{-}\alpha\mathbf{x}_{n\text{-}1}$, assuming that the true states $\mathbf{x}_n$'s in Fig. 4.3.2-A are available to an oracle. As such, we refer to this histogram as the oracle histogram and to the maximum-likelihood GM density fit to these oracle samples as the oracle GM fit in Fig. 4.3.2-C. The constant α is chosen close to and less than one to enforce temporal continuity. To simulate the observed neural response y_t, we use two speech signals of length 90 s each to generate the stimulus vectors required in Eq. (4.34).

We first consider the state-space model of Eq. (4.36) and apply our proposed EM algorithm to the simulated observations to illustrate how the oracle Gaussian mixture fit in Fig. 4.3.2-C can be recovered from observations, and how the multimodal density representation of the process noise can improve TRF estimation under various observation SNRs. We consider $W = 5$, which means that the TRF dynamics are governed by one mixture component of the process noise in windows of length $Wt_0/F_s = 1.5$ s. For simplicity, we consider $\Sigma_{1:M}$ to be diagonal in the parameter set, which makes the update formulas of Eqs. (4.14) and (4.19) to also take diagonal forms. The number of mixture components is chosen as $M = 5$ using the AIC criterion and log-likelihoods computed using Eqs. (4.33) and (4.32). We

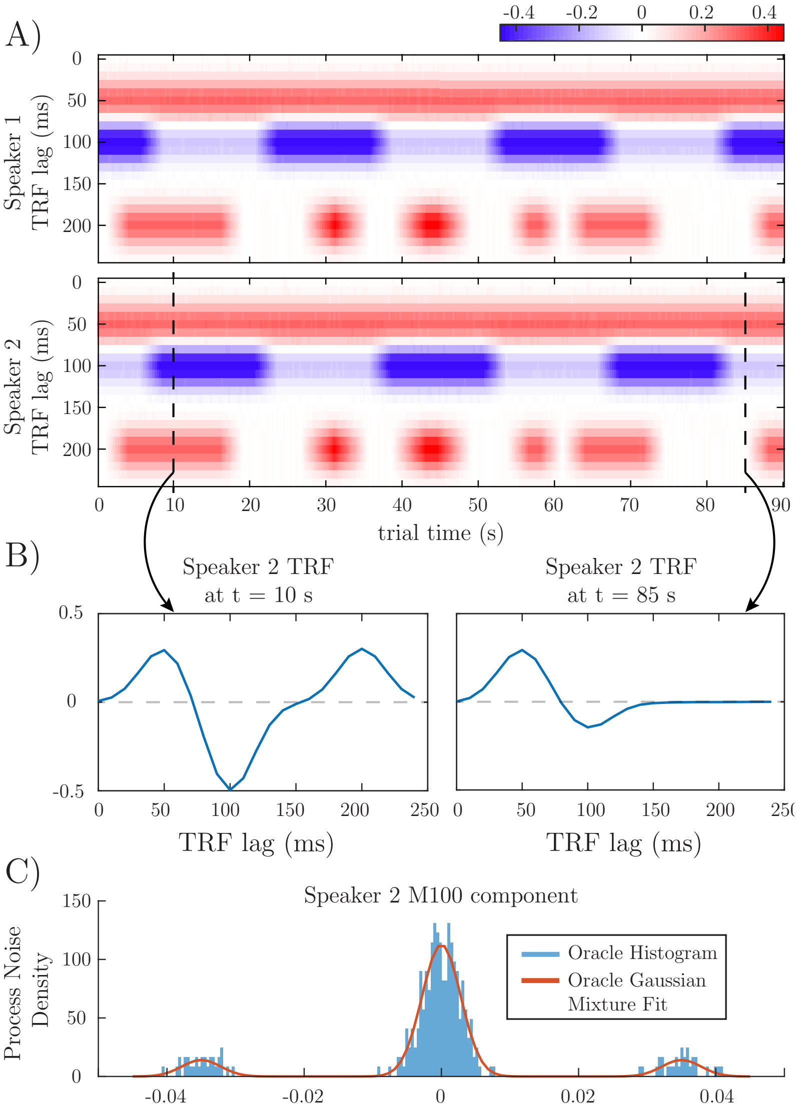

Figure 4.1: Designed simulation study: A) Heatmaps of the synthetic TRFs in time for a two-speaker cocktail party scenario, where the M100 magnitudes are attention-modulated. B) Example instances of speaker 2's TRF when the speaker is attended (left plane) and unattended (right plane). C) Oracle histogram of process noise in (4.35) along the M100 dimension of speaker 2, which is computed from (A), and the fitted GM as the oracle GM fit.

also set $\Gamma_{\mathsf{F}} = \Gamma_{\mathsf{B}} = \Gamma_{\mathsf{S}} = M$. To initialize the EM algorithm, we use two methods: 1) initializing with $\widehat{p}^{(0)}_{1:M} = \frac{1}{M}$, random means $\widehat{\boldsymbol{\mu}}^{(0)}_{1:M}$ close to zero, and $\widehat{\boldsymbol{\Sigma}}^{(0)}_{1:M}$ equal to the estimated process noise covariance in the linear Gaussian model, and 2) setting

$\widehat{\Theta}^{(0)}$ as the GM fit to the empirical samples of process noise in the linear Gaussian model, which are computed from the smoothed state estimates.

Fig. 4.3.2 shows the convergence of the estimated parameters in comparison to those given by the oracle GM fit for a moderate nominal observation SNR of 6.7 dB, using the closed-form approximation approach and the first initialization method. The observation noise variance σ^2 is also estimated within the EM algorithm. The panels for the means and covariances in Fig. 4.3.2 correspond to the 3rd state dimension of speaker 2's TRF (corresponding to the M100 component), in accordance to those in Fig. 4.3.2-C. The mixture probabilities and means of the oracle GM fit are recovered within 30 EM iterations. The covariance elements, however, take a longer time to converge and tend to underestimate those of the oracle GM fit. This shows that at the example nominal SNR of 6.7 dB in our simulation, the algorithm is more sensitive to recovering the average TRF dynamics in each 1.5 s window than to retrieve its detailed variations within the window. It is noteworthy that the initialization points in Fig. 4.3.2-C, given by the estimated process noise variance in a linear Gaussian SSM, are approximately 100 times larger than the variances given by the oracle GM fit. Finally, Fig. 4.3.2-D shows the corresponding estimated process noise density after 200 EM iterations (blue trace), the oracle GM fit (red trace), and the Gaussian model fit obtained from a linear Gaussian SSM used for initialization (yellow trace). While the estimated GM process noise density using our proposed approach closely matches that given by the oracle GM fit, the process noise density obtained by a linear Gaussian model is heavily biased and is not able to capture the multimodal nature of the process. To ease reproducibility, we have archived a

MATLAB implementation of the closed-form approximation method (Approach 2) in the GitHub repository, which reproduces the results of Fig. 4.3.2 [32]. Examples of the convergence curves for the Monte Carlo approximation method (Approach 1) are previously presented in [130], and are omitted here for brevity.

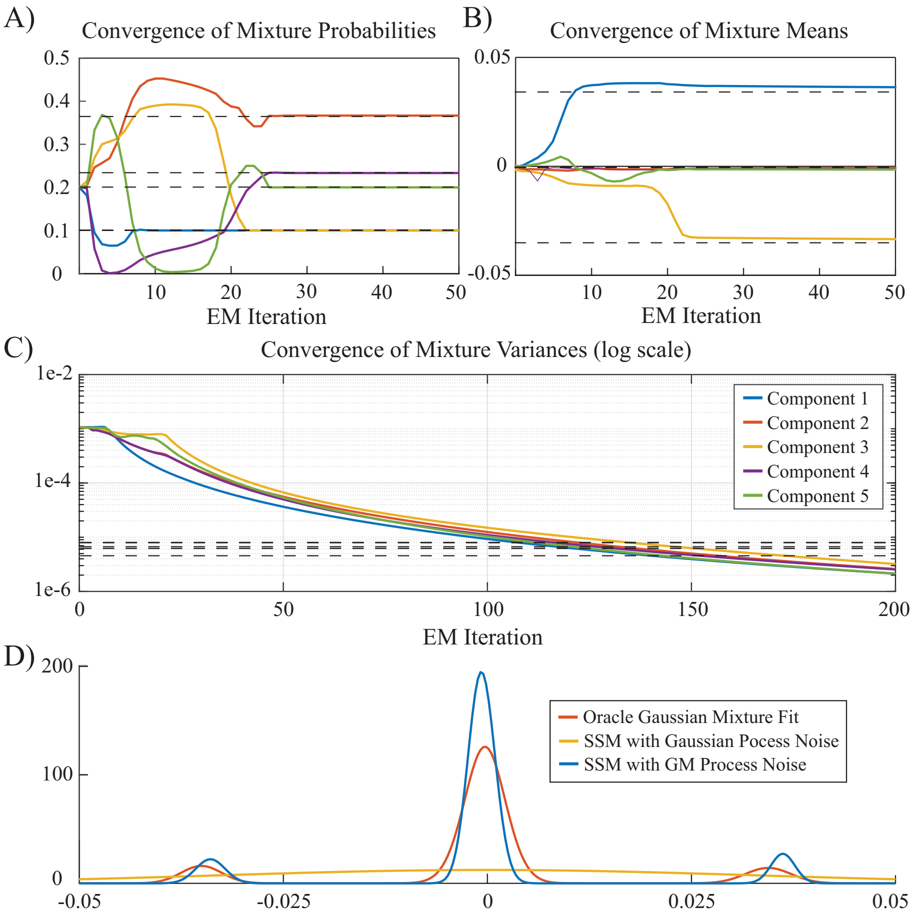

Figure 4.2: Convergence of Gaussian mixture parameters for $M = 5$ in the EM algorithm with closed-form approximations: A) Mixture probabilities. B) Mixture means (along the M100 component of speaker 2 as an example). C) Mixture variances (along the M100 component of speaker 2 as an example). Bold dash lines show the corresponding parameters of the oracle GM fit. D) GM densities (along the M100 component of speaker 2 as an example).

Fig. 4.3.2 shows the normalized RMSE in state estimation with respect to the original states in Fig. 4.3.2-A for nominal observation SNRs in the range [-5.3, 9, 7] dB with 3 dB increments. The results are averaged over 10 runs of the observation noise at each SNR value. For the forgetting factor λ in RLS, an effective estimation length [107] of 2 s is chosen to result in comparable TRF estimates to those of the SSM with $\alpha = 0.99$. Also, γ in Eq. (4.35) for an ℓ_2 penalty is tuned through two-fold cross-validation. For the linear Gaussian SSM and linear SSM with GM process noise in Eq. (4.36), diagonal process noise covariance matrices are considered, and the model parameters and states are estimated simultaneously for each trial run. The SSMs clearly outperform the RLS algorithm in recovering the true states. Also, the SSM with GM process noise with either the closed-form or particle smoothing approximations outperforms the linear Gaussian SSM. We have considered a total of $U = 2000$ particles in Algorithm 7 to approximate densities of dimension $2D(W+1) = 60$ so that state estimates are comparable to those obtained by the closed-form approximation. This resulted in a ten-fold increase in the run-time compared to the closed-form approximation method, which shows the advantage of using the closed-form approximation method. Examples of the estimated TRFs of speaker 1 under the low nominal observation SNR of -5.3 dB are shown in Fig. 4.3.2. The RLS estimate (panel A) exhibits highest variability compared to Fig. 4.3.2-A. While the linear Gaussian SSM estimate in Fig. 4.3.2-B fails to capture the rapid M100 dynamics as well as the steady M50 component (note the M50 and M100 estimates between within the dashed rectangles), the estimate from the SSM with GM process noise in Fig. 4.3.2-C is nearly indistinguishable from the ground truth TRF shown in Fig. 4.3.2-A.

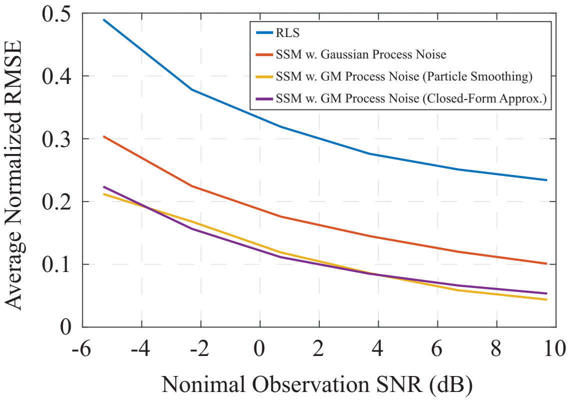

Figure 4.3: Averaged normalized RMSE in state estimation computed over 10 runs of observation noise at each SNR value for dynamic TRF estimation algorithms of: regularized recursive least squares (RLS), linear Gaussian SSM, and linear SSM with GM process noise using closed-form and Monte Carlo particle smoothing approximations. States and noise parameters are both estimated simultaneously from the observations in each run.

4.3.3 Application to Experimentally-Recorded MEG Data

The data used in this work is a subset of recordings in [108] for an at-will attention switching experiment. The participants included five younger-adult (22-33 years old) native English speakers with normal hearing, who were recruited from the University of Maryland. All protocols and procedures were approved by the Institutional Review Board of the university, and written informed consent was obtained from participants.

Experiment Details: Two stories were presented diotically to subjects' ears, one narrated by a male speaker and the other one by a female speaker. The stimuli consisted of two segments from the book, The Legend of Sleepy Hollow by Washington Irving. Subjects listened to three 90 s-long trials of the same speech mixture and were instructed to start attending to the male speaker first, and then to switch

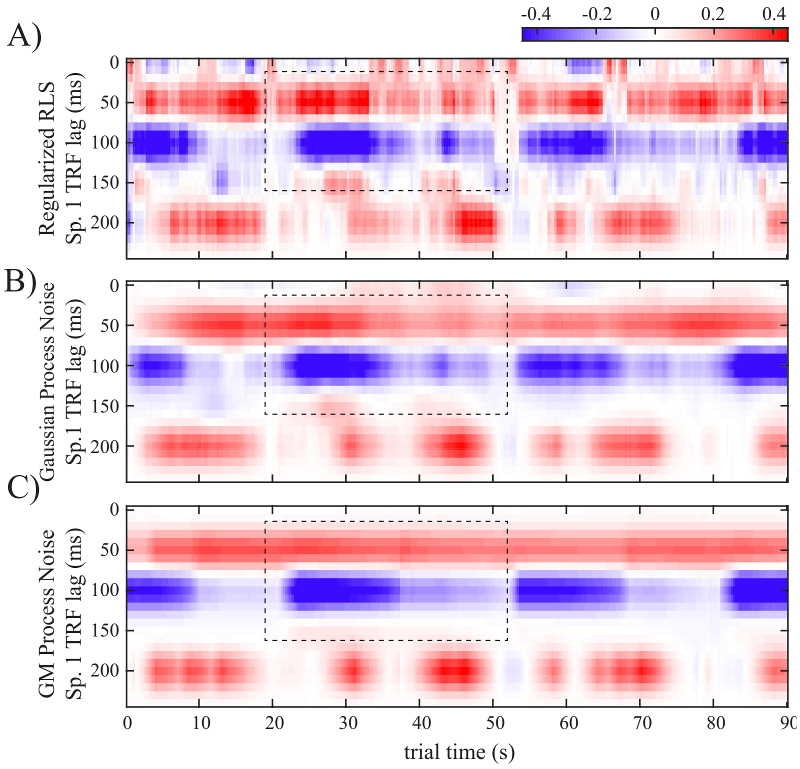

Figure 4.4: Example dynamic TRF estimates for speaker 1 under the low nominal observation SNR of −5.3 dB: A) RLS algorithm. B) Linear Gaussian SSM. C) Linear SSM with GM process noise. The dashed rectangles highlight the differences of these estimates for the sake of comparison.

their attention between the two speakers at their own will for a minimum of one and a maximum of three times during each trial. Subjects were also given a switching button that they were instructed to press every time they decided to switch attention. Prior to the experiment, a single-speaker pilot study was performed where subjects listened to three 60 s-long trials with similar stimuli. Neuromagnetic signals were recorded at a sampling frequency of 2 kHz using a 157-sensor whole-head MEG system (Kanazawa Institute of Technology, Nonoichi Ishikawa, Japan) in a dim magnetically shielded room.

Preprocessings: Three reference channels were used to measure and cancel the environmental magnetic field by using time-shift PCA [131]. All MEG channels were band-pass filtered between 2 Hz and 8 Hz (delta and theta bands), corresponding to the slow temporal modulations in speech [48,106], and downsampled to $F_s = 100$ Hz. In [56], the Denoising Source Separation (DSS) algorithm is described to decompose the MEG data into temporally uncorrelated components ordered according to their trial-to-trial phase-locking reliability. Similar to [51, 107], we consider the first DSS component as the auditory neural response. Thus, we apply the DSS algorithm on pilot trials to compute the subject-specific linear combination of MEG channels that compose the first DSS component. The computed channel maps are then applied to the recorded MEG from the main experiment trials to extract the auditory response y_t in Eq. (4.34). Speech envelopes were similarly filtered and downsampled.

TRF Estimation Results: We set the TRF length to 0.3 s and consider TRFs to be piece-wise constant over windows of length 0.4 s. Also, we choose $W = 5$ to enforce homogeneous TRF dynamics over windows of length 2 s. We represent the TRFs over a Gaussian dictionary with means separated by 20 ms starting from 0 to 280 ms, and variances of $8.5e - 3$. The parameters λ and α are set to 0.92 and 0.97, respectively, to achieve comparable TRF estimates from Eqs. (4.35) and (4.36). The ℓ_2 penalty γ in (4.35) is determined via two-fold cross-validation. We consider diagonal covariance matrices for the process noise to reduce the size of Θ, also estimate the observation noise σ^2 in the EM framework. The forgetting factor mechanism of Eq. (4.35) enforces a temporal continuity in TRF estimates over time and increases robustness to noise and artifacts. The same effect can be replicated in the SSM of Eq. (4.36) by considering α close to one and restricting the dynamic range of the process noise $\mathbf{w}_n$. To enforce the latter, we consider Inverse Gamma (IG) conjugate priors [132] on the diagonal elements of the process noise covariance matrices. For the linear Gaussian SSM with $\mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{Q} = \operatorname{diag}([q_1, \ldots, q_{2D}])$, the log-prior takes the form

$$\kappa \log \mathrm{P}(\mathbf{Q}) = \text{-}\kappa \textstyle\sum_{d=1}^{2D} \Big((\widetilde{\alpha}_d + 1) \log q_d + \widetilde{\beta}_d / q_d \Big) + \mathsf{cst}. \tag{4.37}$$

where $\widetilde{\alpha}_d$ and $\widetilde{\beta}_d$ are the parameters of the IG prior for dimension d and cst. includes terms not dependent on q_d's. The log-prior is then added to the surrogate Q-function of the EM algorithm, and κ determines the strength of the prior with respect to the complete data log-likelihood. We choose $\kappa = N$ for the linear Gaussian case and $\kappa = N/M$ for the linear SSM with GM process noise, to correct for the number of mixture components. We tune the IG parameters using empirical samples of the process noise from the RLS estimates, computed as $\widehat{\mathbf{w}}_n = \widehat{\mathbf{x}}_n^{(\mathsf{RLS})} - \alpha \widehat{\mathbf{x}}_{n\text{-}1}^{(\mathsf{RLS})}$. Thus, the process noise variance is controlled by the IG priors, which prohibit drastic temporal changes in the TRF. For the linear SSM with GM process noise, we also bound the elements of $\widehat{\boldsymbol{\mu}}_{1:M}^{(\ell)}$ in each EM iteration such that the variance of the estimated GM process noise along each dimension is not larger than those of the linear Gaussian case, i.e., estimated q_d's using the EM algorithm. Note that in the absence of such strong priors, the EM algorithm would likely overfit the observed data, resulting in TRFs that are highly variable in time and with no meaningful morphological structure. In our simulation study, the usage of such priors was not necessary, as the $\mathbf{y}_t$'s were directly generated from Eq. (4.34).

Fig. 4.3.3 shows example TRF estimates for two representative trials of one subject. The vertical dashed lines mark reported attention switches by the subject. The number of mixtures M was set to 3 for trial one and 4 for trial two, using the AIC criterion. Row A shows speaker 1's TRF estimate using RLS, which exhibits the highest variability. Rows B and C show the TRF for the linear Gaussian SSM and the linear SSM with GM process noise and inferred using the closed-form approximation, respectively. Although the estimated process noise variance in in the GM case is controlled by that of the Gaussian case in each dimension, we observe that the estimates in row C clearly delineate the heterogeneity of the dynamics of the various TRF components, which are blurred by the linear Gaussian SSM estimates of row B. In other words, the multimodal representation of the process noise allows the model to adapt to rapid changes goverbed by the subjects' behavior. Row D displays speaker 2's TRF estimate using the linear SSM with GM process noise. Comparing rows C and D, we observe the aforementioned attention modulation effect in the magnitude of the M100 components. To illustrate this effect further, row E shows the difference between the M100 magnitudes of the TRFs of speakers 1 and 2, where we locate the M100 at each time as the smallest TRF elements in the $[0.1, 0.2]$ s lag interval. Thus, when speaker 1 (2) is attended, we expect this difference to be positive (negative). The M100 differences for the RLS exhibit high variability (blue traces), and result in inconsistencies with the reported attended speakers (e.g., trial 1 after the 35 s mark, downward arrow). The M100 differences obtained by the linear Gaussian SSM estimates seem to overly smooth those of the RLS (e.g., trial 2, near the 10 s mark, downward arrow). The M100 differences obtained from the proposed linear SSM with GM process noise, however, provide a desirable compromise between these two extremes: Compared to the linear Gaussian SSM, the M100 differences benefit from the clearly delineated TRF dynamics and can result in earlier detection of an attention switch, leading to higher attention decoding accuracy. Instances of this advantage are marked by green arrows in row E, for both trials.

4.4 Concluding Remarks

We considered a SSM with GM process noise and devised an EM algorithm to estimate the parameters of the GM density from SSM observations. To approximate the intractable expectations in EM, we considered two approaches, one based on particle smoothing and another based on closed-form GM approximations to the smoothing densities. As an example application, we considered the problem of dynamic TRF estimation auditory neural responses to speech. We formulated the problem as a linear SSM with Gaussian or GM process noise, and compared the TRF estimates to those of the RLS algorithm used in [107]. Application to simulated data shows that the algorithm can effectively recover the parameters of the underlying GM process noise and that the GM representation improves state estimation for a synthesized latent process exhibiting heterogeneous and rapid dynamics. Application to experimentally-recorded MEG in an at-will attention switching two-speaker cocktail party setting revealed that the proposed SSM with GM process noise model and inference methodology clearly delineates the heterogeneous dynamics of the TRF components that are otherwise not captured by the other techniques. While the proposed methodology can be used as a reliable estimation technique in

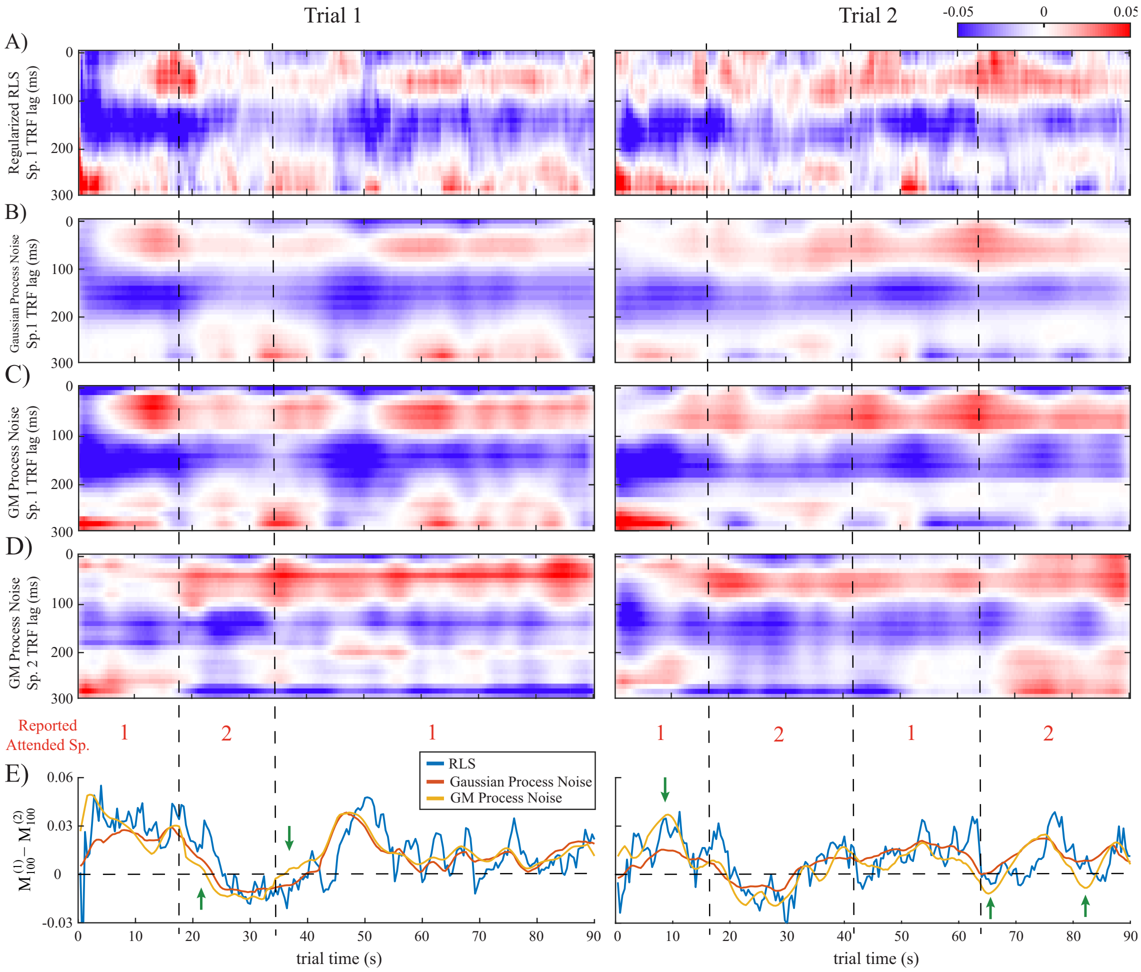

Figure 4.5: TRF estimates for two example trials in an at-will attention switching experiment with vertical dashed lines showing the reported times of attention switches by the subject: A) RLS estimate (speaker 1 TRF). B) Gaussian SSM (speaker 1 TRF). C) SSM with GM process noise (speaker 1 TRF). D) SSM with GM process noise (speaker 2 TRF). E) M100 magnitude differences between the TRFs of speaker 1 and 2 for the different methods. The SSM with GM process noise clearly delineates the heterogeneity of the TRF dynamics and is more consistent with the subjects' behavioral reports (see green arrows), while the RLS estimate is highly variable and the estimate of the Gaussian SSM is overly smooth.

auditory attention decoding applications for the emerging hearing aid technologies, it can be applied to a wider variety of biological problems in which the underlying model exhibits heterogeneous and switching dynamics.

Chapter 5: Conclusion and Future Work

In this thesis we have studied three neural inverse problems: 1) Sparse spectral estimation for neural spiking data, 2) Real-time auditory attention decoding in dual-speaker environments using M/EEG, and 3) Application of state-space models with a Gaussian mixture process noise to dynamic TRF estimation in auditory neuroscience. These problems share the following four key challenges to different extents: First, neuroimaging data is relatively high-dimensional as it includes recordings from hundreds of sensors over potentially large periods of time. However, the underlying neural activity in controlled experiments is often focal, sparse, and structured to some extent either in time, frequency, or spatial domains, or a combination thereof. A major challenge is, therefore, to locate the domain with such characteristics and exploit the structured representation to harness the high-dimensionality. Second, most existing methods in computational neuroscience involve heavy usage of linear models with Gaussian statistics due to their interpretability and convenient estimation. However, to understand the highly complex brain function, we need to move beyond such assumptions and devise efficient estimation algorithms for the relevant nonlinear and non-Gaussian models. Third, with the emergence of advanced BCI systems and neural prosthetics, it is required to develop low-complexity algorithms for analyzing neural activity in a dynamic fashion or in real-time. Finally, efficient, interpretable, and task-specific mathematical representations for neural datasets have to be devised, which can be adopted for diagnosis or soft-decision making.

Having in mind the foregoing challenges, we developed specific algorithms for the three discussed neural inverse problems which extract the sparse spectral profile of neural spiking data, perform near real-time auditory attention decoding using a minimal amount of training data, and provide a comprehensive dynamic analysis of TRFs with rapid tracking of TRF variations. However, each of these methods include specific limitations, which can be overcome either with further algorithm development or design of more precise neuroimaging techniques. For instance, all three of the considered neural inverse problems include non-convex optimization problems, which we have solved using the EM algorithm. EM only guarantees convergence to a local optimum and is, therefore, sensitive to its initialization. We have used the output of simpler models in each task to provide an informed initialization point for EM in our models. However, a comprehensive theoretical understanding of such initializations is lacking, and they can fail to steer the EM algorithm to the global optima or even good-enough local optima. Another example of such limitations is in extracting the auditory component of neural response from M/EEG recordings. In the real-time attention decoding and the dynamic TRF analysis problems, we have adopted the DSS algorithm to compute the auditory portion of the neural response. However, this version of DSS does not exploit the stimulus to extract the auditory response and requires multiple trials of the same experiment to detect the common auditory response and discard the non-persistent parts of the neural response. As a result, DSS is not applicable to dynamic settings or trials with different stimuli, and it can yield responses which are not auditory. Therefore, our dynamic and real-time estimation algorithms can greatly benefit from an algorithm similar to DSS which does not have the discussed shortcomings. Another limitation in dynamic estimation using non-invasive neuroimaging data such as M/EEG is the inherent low SNR of these recordings, which is the main reason why many researchers refrain from moving beyond static and batch-mode estimates using M/EEG. For example, in the auditory attention decoding problem, the computed correlation values for invasive measurements, such as ECoG, are significantly larger than those for M/EEG. As a result, an improvement in non-invasive recording techniques, either in the form of better channel placements or enhanced electrodes, can significantly boost the performance of the devised algorithms for real-time attention decoding and dynamic TRF estimation. As a last constraint, recall that in the real-time attention decoding task, we assumed that the clean speech envelope for each speaker is accessible. However, in practice, these envelopes and other potential speech features have to be extracted from microphone recordings in real-time. Such modules have been recently developed [133], and an interesting future direction can be to combine these modules with the developed algorithm for real-time attention decoding to compare the resulting performances.

Finally, note that the discussed challenges above are shared among many disciplines such as social network analysis, astronomical imaging, and communication systems. Thus, the developed ideas and algorithms in this thesis have potential applications in other domains as well, and existing methods in such disciplines can be leveraged to overcome the foregoing challenges in analyzing neuroimaging data.

Bibliography

[1] D. S. Bassett and E. Bullmore, “Small-world brain networks,” *The neuroscientist*, vol. 12, no. 6, pp. 512–523, 2006.

[2] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann, “Neurophysiological investigation of the basis of the fmri signal,” *Nature*, vol. 412, no. 6843, pp. 150–157, 2001.

[3] G. Buzsaki, *Rhythms of the Brain.* Oxford University Press, 2006.

[4] P. L. Purdon, E. T. Pierce, E. A. Mukamel, M. J. Prerau, J. L. Walsh, K. F. K. Wong, A. F. Salazar-Gomez, P. G. Harrell, A. L. Sampson, A. Cimenser, S. Ching, N. J. Kopell, C. Tavares-Stoeckel, K. Habeeb, R. Merhar, and E. N. Brown, “Electroencephalogram signatures of loss and recovery of consciousness from propofol,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 12, pp. E1142–E1151, 2013.

[5] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, “A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects,” *Journal of neurophysiology*, vol. 93, no. 2, pp. 1074–1089, 2005.

[6] W. Klimesch, “EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis,” *Brain research reviews*, vol. 29, no. 2, pp. 169–195, 1999.

[7] J. A. Pineda, “The functional significance of mu rhythms: translating ”seeing” and ”hearing” into ”doing”,” *Brain Research Reviews*, vol. 50, no. 1, pp. 57–68, 2005.

[8] S. Arroyo and S. Uematsu, “High-frequency eeg activity at the start of seizures.” *Journal of Clinical Neurophysiology*, vol. 9, no. 3, pp. 441–448, 1992.

[9] G. Alarcon, C. Binnie, R. Elwes, and C. Polkey, “Power spectrum and intracranial eeg patterns at seizure onset in partial epilepsy,” *Electroencephalography and clinical neurophysiology*, vol. 94, no. 5, pp. 326–337, 1995.

[10] J. Fell, J. Röschke, K. Mann, and C. Schäffner, "Discrimination of sleep stages: a comparison between spectral and nonlinear EEG measures," *Electroencephalography and clinical Neurophysiology*, vol. 98, no. 5, pp. 401–410, 1996.

[11] A. D. Krystal, J. D. Edinger, W. K. Wohlgemuth, and G. R. Marsh, "Nrem sleep eeg frequency spectral correlates of sleep complaints in primary insomnia subtypes," *Sleep*, vol. 25, no. 6, pp. 630–640, 2002.

[12] R. P. Vertes and R. W. Stackman, *Electrophysiological recording techniques*. Humana Press, 2011, vol. 54.

[13] W. Truccolo, J. A. Donoghue, L. R. Hochberg, E. N. Eskandar, J. R. Madsen, W. S. Anderson, E. N. Brown, E. Halgren, and S. S. Cash, "Single-neuron dynamics in human focal epilepsy," *Nature neuroscience*, vol. 14, no. 5, pp. 635–641, 2011.

[14] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.

[15] L. Paninski, "Maximum likelihood estimation of cascade point-process neural encoding models," *Network: Computation in Neural Systems*, vol. 15, no. 4, pp. 243–262, 2004.

[16] A. C. Smith and E. N. Brown, "Estimating a state-space model from point process observations," *Neural Computation*, vol. 15, no. 5, pp. 965–991, 2003.

[17] M. Chalk, J. L. Herrero, M. A. Gieselmann, L. S. Delicato, S. Gotthardt, and A. Thiele, "Attention reduces stimulus-driven gamma frequency oscillations and spike field coherence in v1," *Neuron*, vol. 66, no. 1, pp. 114–125, 2010.

[18] B. C. Lewandowski and M. Schmidt, "Short bouts of vocalization induce long-lasting fast gamma oscillations in a sensorimotor nucleus," *The Journal of Neuroscience*, vol. 31, no. 39, pp. 13 936–13 948, 2011.

[19] A. Nini, A. Feingold, H. Slovin, and H. Bergman, "Neurons in the globus pallidus do not show correlated activity in the normal monkey, but phase-locked oscillations appear in the mptp model of parkinsonism," *Journal of neurophysiology*, vol. 74, no. 4, pp. 1800–1805, 1995.

[20] L. D. Lewis, V. S. Weiner, E. A. Mukamel, J. A. Donoghue, E. N. Eskandar, J. R. Madsen, W. S. Anderson, L. R. Hochberg, S. S. Cash, E. N. Brown, and P. L. Purdon, "Rapid fragmentation of neuronal networks at the onset of propofol-induced unconsciousness," *Proceedings of the National Academy of Sciences*, vol. 109, no. 49, pp. E3377–E3386, 2012.

[21] D. Ba, B. Babadi, P. L. Purdon, and E. N. Brown, "Robust spectrotemporal decomposition by iteratively reweighted least squares," *Proceedings of the National Academy of Sciences*, vol. 111, no. 50, pp. E5336–E5345, 2014.

[22] D. B. Percival, *Spectral analysis for physical applications.* Cambridge University Press, 1993.

[23] G. A. Worrell, L. Parish, S. D. Cranstoun, R. Jonas, G. Baltuch, and B. Litt, "High-frequency oscillations and seizure generation in neocortical epilepsy," *Brain*, vol. 127, no. 7, pp. 1496–1506, 2004.

[24] R. Barbieri and E. N. Brown, "Analysis of heartbeat dynamics by point process adaptive filtering," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 1, pp. 4–12, 2006.

[25] Z. Chen, E. N. Brown, and R. Barbieri, "Assessment of autonomic control and respiratory sinus arrhythmia using point process models of human heart beat dynamics," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 7, pp. 1791–1802, 2009.

[26] D. Vere-Jones, "An introduction to the theory of point processes," *Springer Ser. Statist., Springer, New York*, 1988.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning.* Springer, 2009.

[29] D. M. Halliday and J. R. Rosenberg, "Time and frequency domain analysis of spike train and time series data," in *Modern techniques in neuroscience research.* Springer, 1999, pp. 503–543.

[30] D. J. Thomson, "Spectrum estimation and harmonic analysis," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1055–1096, 1982.

[31] B. Babadi and E. N. Brown, "A review of multitaper spectral analysis." *Biomedical Engineering, IEEE Transactions on*, vol. 61, no. 5, pp. 1555–1564, 2014.

[32] S. Miran, *Real-Time Tracking of Selective Auditory Attention MATLAB Code.* Available on GitHub Repository: https://github.com/sinamiran/Real-Time-Tracking-of-Selective-Auditory-Attention, 2017.

[33] L. D. Lewis, S. Ching, V. S. Weiner, R. A. Peterfreund, E. N. Eskandar, S. S. Cash, E. N. Brown, and P. L. Purdon, "Local cortical dynamics of burst suppression in the anaesthetized brain," *Brain*, vol. 136, no. 9, pp. 2727–2737, 2013.

[34] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[35] J. M. Tang and Y. Saad, "A probing method for computing the diagonal of a matrix inverse," *Numerical Linear Algebra with Applications*, vol. 19, no. 3, pp. 485–501, 2012.

[36] D. S. Brungart, "Informational and energetic masking effects in the perception of two simultaneous talkers," *The Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1101–1109, 2001.

[37] J. H. McDermott, "The cocktail party problem," *Current Biology*, vol. 19, no. 22, pp. R1024–R1027, 2009.

[38] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[39] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[40] J. C. Middlebrooks, J. Z. Simon, A. N. Popper, and R. R. Fay, *The Auditory System at the Cocktail Party.* in the Springer Handbook of Auditory Research series, 2017.

[41] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[42] T. D. Griffiths and J. D. Warren, "What is an auditory object?" *Nature reviews. Neuroscience*, vol. 5, no. 11, p. 887, 2004.

[43] Y. I. Fishman and M. Steinschneider, "Neural correlates of auditory scene analysis based on inharmonicity in monkey primary auditory cortex," *Journal of Neuroscience*, vol. 30, no. 37, pp. 12 480–12 494, 2010.

[44] S. A. Shamma, M. Elhilali, and C. Micheyl, "Temporal coherence and attention in auditory scene analysis," *Trends in neurosciences*, vol. 34, no. 3, pp. 114–123, 2011.

[45] J. A. O'Sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor, "Attentional selection in a cocktail party environment can be decoded from single-trial eeg," *Cerebral Cortex*, vol. 25, no. 7, pp. 1697–1706, 2015.

[46] A. J. Power, J. J. Foxe, E.-J. Forde, R. B. Reilly, and E. C. Lalor, "At what time is the cocktail party? a late locus of selective attention to natural speech," *European Journal of Neuroscience*, vol. 35, no. 9, pp. 1497–1503, 2012.

[47] B. Mirkovic, S. Debener, M. Jaeger, and M. De Vos, "Decoding the attended speech stream with multi-channel eeg: implications for online, daily-life applications," *Journal of neural engineering*, vol. 12, no. 4, p. 046007, 2015.

[48] N. Ding and J. Z. Simon, "Emergence of neural encoding of auditory objects while listening to competing speakers," *Proceedings of the National Academy of Sciences*, vol. 109, no. 29, pp. 11 854–11 859, 2012.

[49] ——, "Neural coding of continuous speech in auditory cortex during monaural and dichotic listening," *Journal of neurophysiology*, vol. 107, no. 1, pp. 78–89, 2012.

[50] S. Akram, J. Z. Simon, S. A. Shamma, and B. Babadi, "A state-space model for decoding auditory attentional modulation from meg in a competing-speaker environment," in *Advances in Neural Information Processing Systems*, 2014, pp. 460–468.

[51] S. Akram, A. Presacco, J. Z. Simon, S. A. Shamma, and B. Babadi, "Robust decoding of selective auditory attention from meg in a competing-speaker environment via state-space modeling," *NeuroImage*, vol. 124, pp. 906–917, 2016.

[52] S. Akram, J. Z. Simon, and B. Babadi, "Dynamic estimation of the auditory temporal response function from meg in competing-speaker environments," *IEEE Transactions on Biomedical Engineering*, 2016.

[53] E. M. Kaya and M. Elhilali, "Modelling auditory attention," *Phil. Trans. R. Soc. B*, vol. 372, no. 1714, p. 20160101, 2017.

[54] A. Sheikhattar, J. B. Fritz, S. A. Shamma, and B. Babadi, "Adaptive sparse logistic regression with application to neuronal plasticity analysis," in *Signals, Systems and Computers, 2015 49th Asilomar Conference on.* IEEE, 2015, pp. 1551–1555.

[55] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering.* Springer New York, 2011, pp. 185–212.

[56] A. de Cheveigne and J. Z. Simon, "Denoising based on spatial filtering," *Journal of neuroscience methods*, vol. 171, no. 2, pp. 331–339, 2008.

[57] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[58] A. Sheikhattar, J. B. Fritz, S. A. Shamma, and B. Babadi, "Recursive sparse point process regression with application to spectrotemporal receptive field plasticity analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2026–2039, 2015.

[59] W. Biesmans, J. Vanthornhout, J. Wouters, M. Moonen, T. Francart, and A. Bertrand, “Comparison of speech envelope extraction methods for eeg-based auditory attention detection in a cocktail party scenario,” in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE.* IEEE, 2015, pp. 5155–5158.

[60] A. Aroudi, B. Mirkovic, M. De Vos, and S. Doclo, “Auditory attention decoding with eeg recordings using noisy acoustic reference signals,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 694–698.

[61] W. Biesmans, N. Das, T. Francart, and A. Bertrand, “Auditory-inspired speech envelope extraction methods for improved eeg-based auditory attention detection in a cocktail party scenario,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 5, pp. 402–412, 2017.

[62] J. O’Sullivan, Z. Chen, J. Herrero, G. M. McKhann, S. A. Sheth, A. D. Mehta, and N. Mesgarani, “Neural decoding of attentional selection in multi-speaker environments without access to clean sources,” *Journal of Neural Engineering*, vol. 14, no. 5, 2017.

[63] S. Van Eyndhoven, T. Francart, and A. Bertrand, “Eeg-informed attended speaker extraction from recorded speech mixtures with application in neuro-steered hearing prostheses,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 5, pp. 1045–1056, 2017.

[64] B. Babadi, N. Kalouptsidis, and V. Tarokh, “SPARLS: The sparse RLS algorithm,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4013–4025, 2010.

[65] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski, “Proximal methods for sparse hierarchical dictionary learning,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 487–494.

[66] J. Duchi and Y. Singer, “Efficient online and batch learning using forward backward splitting,” *Journal of Machine Learning Research*, vol. 10, no. Dec, pp. 2899–2934, 2009.

[67] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE Journal of selected topics in signal processing*, vol. 1, no. 4, pp. 586–597, 2007.

[68] T. Goldstein, C. Studer, and R. Baraniuk, “A field guide to forward-backward splitting with a fasta implementation,” *arXiv preprint arXiv:1411.3406*, 2014.

[69] ——, “A field guide to forward-backward splitting with a FASTA implementation,” *arXiv eprint*, vol. abs/1411.3406, 2014. [Online]. Available: http://arxiv.org/abs/1411.3406

[70] ——, "FASTA: A generalized implementation of forward-backward splitting," January 2015, http://arxiv.org/abs/1501.04979.

[71] A. C. Smith, L. M. Frank, S. Wirth, M. Yanike, D. Hu, Y. Kubota, A. M. Graybiel, W. A. Suzuki, and E. N. Brown, "Dynamic analysis of learning in behavioral experiments," *Journal of Neuroscience*, vol. 24, no. 2, pp. 447–461, 2004.

[72] M. A. Tanner, *Tools for statistical inference.* Springer, 1991, vol. 3.

[73] P. De Jong and M. J. Mackinnon, "Covariances for smoothed estimates in state space models," *Biometrika*, vol. 75, no. 3, pp. 601–602, 1988.

[74] B. Khalighinejad, G. C. da Silva, and N. Mesgarani, "Dynamic encoding of acoustic features in neural responses to continuous speech," *Journal of Neuroscience*, vol. 37, no. 8, pp. 2176–2185, 2017.

[75] S. Kähkönen, J. Ahveninen, I. P. Jääskeläinen, S. Kaakkola, R. Näätänen, J. Huttunen, and E. Pekkonen, "Effects of haloperidol on selective attention: a combined whole-head meg and high-resolution eeg study," *Neuropsychopharmacology*, vol. 25, no. 4, pp. 498–504, 2001.

[76] M. G. Bleichner, B. Mirkovic, and S. Debener, "Identifying auditory attention with ear-eeg: ceegrid versus high-density cap-eeg comparison," *Journal of neural engineering*, vol. 13, no. 6, p. 066004, 2016.

[77] J. Särelä and H. Valpola, "Denoising source separation," *Journal of machine learning research*, vol. 6, no. Mar, pp. 233–272, 2005.

[78] M. Chait, J. Z. Simon, and D. Poeppel, "Auditory m50 and m100 responses to broadband noise: functional implications," *Neuroreport*, vol. 15, no. 16, pp. 2455–2458, 2004.

[79] M. Chait, A. de Cheveigné, D. Poeppel, and J. Z. Simon, "Neural dynamics of attending and ignoring in human auditory cortex," *Neuropsychologia*, vol. 48, no. 11, pp. 3262–3271, 2010.

[80] B. D. Anderson and J. B. Moore, *Optimal filtering.* Courier Corporation, 2012.

[81] G. C. Goodwin, S. F. Graebe, M. E. Salgado *et al.*, *Control system design.* Prentice Hall New Jersey, 2001, vol. 240.

[82] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 564–575, 2003.

[83] Y. Zeng and S. Wu, *State-space models: Applications in economics and finance.* Springer, 2013, vol. 1.

[84] W. Wu, J. E. Kulkarni, N. G. Hatsopoulos, and L. Paninski, “Neural decoding of hand motion using a linear state-space model with hidden states,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 4, pp. 370–378, 2009.

[85] A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, “Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control,” *Neuron*, vol. 82, no. 6, pp. 1380–1393, 2014.

[86] S. Miran, S. Akram, A. Sheikhattar, J. Z. Simon, T. Zhang, and B. Babadi, “Real-time tracking of selective auditory attention from M/EEG: A Bayesian filtering approach,” *Frontiers in Neuroscience*, vol. 12, 2018.

[87] S. Gibson and B. Ninness, “Robust maximum-likelihood estimation of multivariable dynamic systems,” *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005.

[88] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[89] Z. Ghahramani and G. E. Hinton, “Variational learning for switching state-space models,” *Neural computation*, vol. 12, no. 4, pp. 831–864, 2000.

[90] O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[91] G. Kitagawa, “Non-Gaussian statespace modeling of nonstationary time series,” *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1041, 1987.

[92] G. Kitagawa and W. Gersch, *Smoothness priors analysis of time series*. Springer Science & Business Media, 1996, vol. 116.

[93] H. W. Sorenson and D. L. Alspach, “Recursive bayesian estimation using gaussian sums,” *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[94] G. Kitagawa, “The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother,” *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.

[95] J. H. Kotecha and P. M. Djuric, “Gaussian sum particle filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[96] D. Barber, “Expectation correction for smoothed inference in switching linear dynamical systems,” *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2515–2540, 2006.

[97] B.-N. Vo, B.-T. Vo, and R. P. Mahler, "Closed-form solutions to forward–backward smoothing," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2012.

[98] J. Lo, "Finite-dimensional sensor orbits and optimal nonlinear filtering," *IEEE Transactions on Information Theory*, vol. 18, no. 5, pp. 583–588, 1972.

[99] C.-B. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 418–425, 1978.

[100] C.-J. Kim, "Dynamic linear models with markov-switching," *Journal of Econometrics*, vol. 60, no. 1-2, pp. 1–22, 1994.

[101] L. Blackmore, S. Gil, S. Chung, and B. Williams, "Model learning for switching linear systems with autonomous mode transitions," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 4648–4655.

[102] A. Svensson, T. B. Schön, and F. Lindsten, "Identification of jump markov linear models using particle filters," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6504–6509.

[103] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. M. Dekker New York, 1988, vol. 84.

[104] J. Olsson, O. Cappé, R. Douc, E. Moulines *et al.*, "Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models," *Bernoulli*, vol. 14, no. 1, pp. 155–179, 2008.

[105] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[106] N. Ding and J. Z. Simon, "Neural coding of continuous speech in auditory cortex during monaural and dichotic listening," *American Journal of Physiology-Heart and Circulatory Physiology*, 2011.

[107] S. Akram, J. Z. Simon, and B. Babadi, "Dynamic estimation of the auditory temporal response function from MEG in competing-speaker environments," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1896–1905, 2017.

[108] A. Presacco, S. Miran, B. Babadi, and J. Z. Simon, "Real-time tracking of magnetoencephalographic neuromarkers during a dynamic attention-switching task," in *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019.

[109] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models," *Computational Statistics & Data Analysis*, vol. 41, no. 3-4, pp. 561–575, 2003.

[110] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan, "Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences," in *Advances in neural information processing systems*, 2016, pp. 4116–4124.

[111] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, Tech. Rep., 1996.

[112] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If I had a million particles," in *Proceedings of the 23rd International Conference on Machine Learning.* ACM, 2006, pp. 481–488.

[113] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[114] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373).* Ieee, 2000, pp. 153–158.

[115] R. Wong, *Asymptotic approximations of integrals.* SIAM, 2001, vol. 34.

[116] A. R. Runnalls, "Kullback-leibler approach to gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, 2007.

[117] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at gaussian mixture reduction algorithms," in *14th International Conference on Information Fusion.* IEEE, 2011, pp. 1–8.

[118] J. Ding, V. Tarokh, and Y. Yang, "Model selection techniques: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 6, pp. 16–34, 2018.

[119] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 381–396, 2002.

[120] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[121] A. Mehrjou, R. Hosseini, and B. N. Araabi, "Improved bayesian information criterion for mixture model selection," *Pattern Recognition Letters*, vol. 69, pp. 22–27, 2016.

[122] P. De Jong, "The likelihood for a state space model," *Biometrika*, vol. 75, no. 1, pp. 165–169, 1988.

[123] A. de Cheveigné, D. D. Wong, G. M. Di Liberto, J. Hjortkjaer, M. Slaney, and E. Lalor, "Decoding the auditory brain with canonical component analysis," *NeuroImage*, vol. 172, pp. 206–216, 2018.

[124] G. M. Di Liberto, J. A. OSullivan, and E. C. Lalor, "Low-frequency cortical entrainment to speech reflects phoneme-level processing," *Current Biology*, vol. 25, no. 19, pp. 2457–2465, 2015.

[125] M. P. Broderick, A. J. Anderson, G. M. Di Liberto, M. J. Crosse, and E. C. Lalor, "Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech," *Current Biology*, vol. 28, no. 5, pp. 803–809, 2018.

[126] C. Brodbeck, A. Presacco, and J. Z. Simon, "Neural source dynamics of brain responses to continuous stimuli: Speech processing from acoustics to comprehension," *NeuroImage*, vol. 172, pp. 162–174, 2018.

[127] D. D. Wong, S. Fuglsang, J. Hjortkjær, E. Ceolini, M. Slaney, and A. de Cheveigné, "A comparison of temporal response function estimation methods for auditory attention decoding," 2018.

[128] N. Kalouptsidis and S. Theodoridis, *Adaptive system identification and signal processing algorithms.* Prentice Hall New York, 1993, vol. 994.

[129] L. Ljung, "General structure of adaptive algorithms: adaptation and tracking," 1991.

[130] S. Miran, J. Z. Simon, M. C. Fu, S. I. Marcus, and B. Babadi, "Estimation of state-space models with gaussian mixture process noise," in *2nd IEEE Data Science Workshop.* IEEE, 2019.

[131] A. De Cheveigné and J. Z. Simon, "Denoising based on time-shift pca," *Journal of neuroscience methods*, vol. 165, no. 2, pp. 297–305, 2007.

[132] P. Diaconis and D. Ylvisaker, "Conjugate priors for exponential families," *The Annals of statistics*, pp. 269–281, 1979.

[133] C. Han, J. OSullivan, Y. Luo, J. Herrero, A. D. Mehta, and N. Mesgarani, "Speaker-independent auditory attention decoding without access to clean speech sources," *Science advances*, vol. 5, no. 5, p. eaav6134, 2019.