Algorithms for Estimating Time-Locked Neural Response Components in Cortical Processing of Continuous Speech

Joshua P. Kulasingham and Jonathan Z. Simon

Abstract— Objective: **The Temporal Response Function (TRF) is a linear model of neural activity time-locked to continuous stimuli, including continuous speech. TRFs based on speech envelopes typically have distinct components that have provided remarkable insights into the cortical processing of speech. However, current methods may lead to less than reliable estimates of single-subject TRF components. Here, we compare two established methods, in TRF component estimation, and also propose novel algorithms that utilize prior knowledge of these components, bypassing the full TRF estimation.** ***Methods:*** **We compared two established algorithms, ridge and boosting, and two novel algorithms based on Subspace Pursuit (SP) and Expectation Maximization (EM), which directly estimate TRF components given plausible assumptions regarding component characteristics. Single-channel, multi-channel, and source-localized TRFs were fit on simulations and real magnetoencephalographic data. Performance metrics included model fit and component estimation accuracy.** ***Results:*** **Boosting and ridge have comparable performance in component estimation. The novel algorithms outperformed the others in simulations, but not on real data, possibly due to the plausible assumptions not actually being met. Ridge had slightly better model fits on real data compared to boosting, but also more spurious TRF activity.** ***Conclusion:*** **Results indicate that both smooth (ridge) and sparse (boosting) algorithms perform comparably at TRF component estimation. The SP and EM algorithms may be accurate, but rely on assumptions of component characteristics.** ***Significance:*** **This systematic comparison establishes the suitability of widely used and novel algorithms for estimating robust TRF components, which is essential for improved subject-specific investigations into the cortical processing of speech.**

Index Terms — **MEG, EEG, auditory, deconvolution, reverse correlation, attention, cocktail party, matching pursuit, ERP**

I. Introduction

THE human brain time-locks to features of continuous speech, extracting meaningful information relevant to comprehension. Magnetoencephalography (MEG) and electroencephalography (EEG) are suitable methods to measure these time-locked responses, due to their high temporal resolution. Traditional methods for analyzing auditory responses involve averaging over multiple trials of repeated stimuli to estimate Evoked Response Potentials (ERPs) [1], [2]. But exploring the complex mechanisms involved in speech processing requires non-repetitive, continuous speech stimuli of long duration, and averaging over trials is no longer feasible. One method of analyzing responses to continuous stimuli uses linear models called Temporal Response Functions (TRFs), that estimate the impulse response of the neural system to continuous stimuli [3], [4]. TRFs based

This work was supported by the National Science Foundation (SMA-1734892), and the National Institutes of Health (R01-DC014085 and R01-DC019394).

J. P. Kulasingham is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA. (joshuapk@terpmail.umd.edu)

J. Z. Simon is with the Department of Electrical and Computer Engineering, the Institute for Systems Research, and the Department of Biology, University of Maryland, College Park, MD, USA.

on neural recordings using magnetoencephalography (MEG) have response components such as the M50 (~50 ms latency), M100 (~100-150 ms) and M200 (~200-250 ms) that are analogous to well-known auditory ERP components, the P1, N1, and P2 components of electroencephalography (EEG), and which have been utilized to investigate selective attention [3], [5]–[7], linguistic processing [8]–[10], and age-related differences in the auditory system [11]. However, though estimated TRFs display these canonical components at the group-average level, individual TRFs are much noisier and do not always have well-defined components. It is essential to detect robust response components on a per-subject level, both to identify task effects and for biomedical applications such as smart hearing aids. Hence, the suitability of various TRF methods for component estimation must be determined.

Variations of regularized regression and machine learning methods for estimating TRFs have been previously compared for decoding subject attention in a multi-talker scenario [6], [12], [13]. However, it is unclear how they compare to commonly used sparse TRF estimation techniques such as boosting [14], [15]. Furthermore, a focus on model fits for attention decoding may not be suitable for studies interested in accurate estimation of TRF components.

In this work we perform a systematic comparison of TRF algorithms in terms of estimating TRF components. Two widely used TRF estimation algorithms are ridge regression [13], [16] and boosting [3], [14], [15]. The former uses ℓ_2 regularization which leads to smooth TRFs with broad components, while the latter greedily adds values to the TRF, thereby prioritizing sparsity in the TRF and leading to narrower, sharper components. However, it is not clear which of these methods is more accurate in estimating TRF component latencies and amplitudes.

Both ridge and boosting do not place restrictions on the number or latencies of specific TRF components. Since canonical auditory response components are often present in TRFs to the speech envelope, it is reasonable to incorporate this information during estimation. Several methods have been proposed for directly estimating latencies and amplitudes for M/EEG evoked responses (but not for TRFs). The earliest ERP latency estimation methods involved cross correlation with average response templates [17]. More recent algorithms have utilized techniques such as Independent Component Analysis [18], [19], wavelet decomposition [20], maximum likelihood estimation [21], [22], autoregressive models [23], Expectation Maximization (EM) [24], Matching Pursuit [25] and Bayesian methods [26], [27].

In this work, we propose novel TRF component estimation algorithms that utilize prior knowledge of the characteristics of neural responses (i.e., component latency ranges), and directly estimate component latencies, amplitudes and topographies. The first proposed algorithm estimates single-channel TRF component latencies and amplitudes using Subspace Pursuit (SP) [28]. The second algorithm extends this method for multi-channel TRFs using SP and Expectation Maximization (EM) [24], [29], and also directly estimates sensor topographies or cortical source distributions of TRF components. The SP algorithm is widely used for sparse signal recovery and is typically capable of recovering components in an efficient manner. The EM algorithm is a maximum likelihood method that is able to incorporate 'hidden' variables and is widely used in signal estimation [30]. Pursuit algorithms and EM have been used for single trial evoked response estimation [24], [25], and here, we employ natural extensions of these algorithms for TRF component estimation.

A simulation study, and an application of these algorithms to a real dataset, are reported and their performance is compared using single-channel, multi-channel, and source localized TRFs. Performance metrics include the correlation between the actual and the predicted signal, which is the conventional measure of model fit, and several other measures of component estimation accuracy. Throughout this work, "model fit" denotes the Pearson correlation between the actual and predicted signals. Other considerations such as spurious TRF activity and missing components are also examined. In summary, this work discusses the strengths and weaknesses of widely used algorithms and proposes novel methods for TRF component estimation that may provide robust and interpretable time-locked response components.

II. Methods

A. Established Algorithms for TRF estimation

The TRF estimation problem is given by the convolution

$$\mathbf{y} = \boldsymbol{\beta} * \mathbf{x} + \mathbf{n} \qquad (1)$$

Where $\mathbf{y} \in \mathbb{R}^T$ is the vector of the single-channel measured signal (e.g., at one sensor) for T time points, $\mathbf{x} \in \mathbb{R}^T$ is the predictor variable (e.g., the speech envelope), $\boldsymbol{\beta} \in \mathbb{R}^K$ is the corresponding TRF over K time lags, and $\mathbf{n} \in \mathbb{R}^T$ is the noise. This can be reformulated as a regression as follows

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{n} \qquad (2)$$

Where $\mathbf{X} \in \mathbb{R}^{T\times K}$ is the Toeplitz matrix formed by lagged predictor values. The well-known ridge regression algorithm has been widely used to solve this problem [16]. Another commonly used technique is the boosting algorithm, which is a sparse estimation technique belonging to the broad family of greedy additive estimators, and solves the TRF problem using coordinate descent [14], [15], [31]. In brief, this algorithm starts from an all-zero TRF and incrementally adds small, fixed values to the TRF to decrease the mean square error (MSE) at each iteration. The iterations are stopped when the MSE does not improve. A dictionary of basis elements (e.g., Hamming windows) is used for the incremental additions to the TRF. Both ridge and boosting can be used independently at each sensor to estimate TRFs for multi-channel data.

B. Proposed SP algorithm for TRF estimation

The SP algorithm searches for TRF components within predefined latency windows and directly estimates them. This is unlike the ridge and boosting algorithms that do not place specific restrictions on the number or latencies of detected TRF components. Assuming there are J components (e.g., $J = 3$ for M50, M100, M200 components), the TRF model is now given by a modified version of (1).

$$\mathbf{y} = \textstyle\sum_{j=1}^{J} a_j \mathbf{X}\mathbf{c}_j + \mathbf{n} \qquad (3)$$

Where $a_j \in \mathbb{R}$ and $\mathbf{c}_j \in \mathbb{R}^K$ are the amplitude and waveform for the j^{th} component. The component waveforms $\mathbf{c}_j$ are selected according to the component latency τ_j from a basis dictionary (e.g., Hamming windows) that span the TRF lags (i.e., $\mathbf{c}_j$ is column number τ_j of the basis dictionary matrix). The SP algorithm directly estimates the amplitudes a_j and latencies τ_j. The complete algorithm is given in Algorithm 1.

Algorithm 1: SP for TRF estimation

```
Inputs: Measured signal $\mathbf{y} \epsilon \mathbb{R}^T$, predictor matrix $\mathbf{X} \epsilon \mathbb{R}^{T \times K}$, number of components $J$ and corresponding latency windows $W_j$
Initialize the set of TRF components to the empty set; $\mathcal{C}^0 = \emptyset$.
Set the residual to the measured signal $\mathbf{r}^0 = \mathbf{y}$
repeat for $l = 1,2, \dots$
    repeat for $j = 1, .., J$
        Find the best component latency
            $\mathbf{c}_j^* = \underset{\tau \in W_j}{\operatorname{argmax}} |< \mathbf{r}^l, \mathbf{X}\mathbf{c}_\tau >|$
        where $\mathbf{c}_\tau$ is the basis component with latency $\tau$
    Add the $J$ new components to the set $\tilde{\mathcal{C}} = \mathcal{C}^{l-1} \cup \{\mathbf{c}_j^*\}$
    Estimate amplitudes $\tilde{\boldsymbol{a}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$
        where $\mathbf{A}$ has columns $\{\mathbf{Xc} \mid \mathbf{c} \in \tilde{\mathcal{C}}\}$
    Update the component set
     $\mathcal{C}^l = \{J$ components with the largest amplitudes for each $W_j\}$
    Re-estimate amplitudes $\boldsymbol{a}^l = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y}$
        where $\mathbf{B}$ has columns $\{\mathbf{Xc} \mid \mathbf{c} \in \mathcal{C}^l\}$
    Calculate the new residual $\mathbf{r}^l = \mathbf{y} - \mathbf{B}\boldsymbol{a}^l$
    If $\|\mathbf{r}^l\| > \|\mathbf{r}^{l-1}\|$ stop iterations and set $\mathcal{C}^l = \mathcal{C}^{l-1}$ & $\boldsymbol{a}^l = \boldsymbol{a}^{l-1}$
Output: amplitudes $\boldsymbol{a}^l = [a_1, \dots, a_J]$, components $\mathbf{c}_j \in \mathcal{C}^l$ and TRF $\boldsymbol{\beta} = \sum_{j=1}^J a_j\mathbf{c}_j$.
```

The SP algorithm estimates TRFs composed of only the required number of components, and can also be applied independently at each sensor for multi-channel TRFs.

C. *Proposed EM-SP algorithm for TRF Estimation*

The EM-SP algorithm is an extension of the SP algorithm for multidimensional TRFs. In addition to directly estimating amplitudes and latencies, this algorithm also directly estimates sensor topographies or source distributions for multi-channel TRFs. This algorithm uses EM to iteratively estimate component topographies in the E-step, and latencies using SP in the M-step. Given a predefined number of components and corresponding latency windows, the EM-SP multi-channel TRF model is given by a modified version of (3).

$$\mathbf{Y} = \sum_j \mathbf{z}_j \left(\mathbf{X}\mathbf{c}_j\right)^{\mathrm{T}} + \mathbf{N} \tag{4}$$

Where $\mathbf{Y} \epsilon \mathbb{R}^{M \times T}$ is the measured data over M sensors and T time points, $\mathbf{z}_j \epsilon \mathbb{R}^M$ is the spatial topography of the j^{th} component, $\mathbf{c}_j \epsilon \mathbb{R}^K$ is the temporal waveform of the j^{th} component, $\mathbf{X} \epsilon \mathbb{R}^{T \times K}$ is the predictor matrix, and $\mathbf{N} \epsilon \mathbb{R}^{M \times T}$ is the measurement noise. The component latency is given by τ_j and is related to (4) by the fact that $\boldsymbol{c}_j$ corresponds to column number τ_j in the TRF basis dictionary matrix. We assume the following priors,

$$\begin{gathered} \mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R}) \\ \mathbf{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{T \times T} \otimes \boldsymbol{\Lambda}) \end{gathered} \tag{5}$$

Where the temporal noise covariance is assumed to be the identity matrix and the spatial noise covariance is given by $\boldsymbol{\Lambda} \epsilon \mathbb{R}^{M \times M}$. For the EM algorithm, we consider the spatial topographies $\mathcal{Z} = \{\mathbf{z}_j\}$ as the 'hidden' variables. The remaining parameters that need to be estimated are $\Theta = \left\{ \boldsymbol{\mu}, \mathbf{R}, \boldsymbol{\Lambda}, \tau_j \right\}$. Detailed derivations of the algorithm are provided in supplementary materials. Here, we summarize the main steps of the algorithm.

The Q-function is given by taking the expectation over the posterior probability $p(\mathcal{Z}|\mathbf{Y}, \Theta)$.

$$Q\left(\Theta|\Theta^{(t)}\right) = \frac{T}{2}\log|\boldsymbol{\Lambda}^{-1}| + \frac{J}{2}\log|\mathbf{R}^{-1}| - \frac{1}{2}tr[\mathbf{Y}^T\boldsymbol{\Lambda}^{-1}\mathbf{Y}] + tr\left[\mathbf{Y}^T\boldsymbol{\Lambda}^{-1}\left(\sum_j \mathbf{E}[\mathbf{z}_j]\mathbf{x}_j^T\right)\right] - \frac{1}{2}\, tr\left[\sum_i \sum_j \mathbf{x}_j^T \mathbf{x}_i \mathbf{E}\left[\mathbf{z}_j \mathbf{z}_i^T\right] \boldsymbol{\Lambda}^{-1}\right] \\ -\frac{1}{2}\sum_j tr\left(\mathbf{E}\left[\mathbf{z}_j\mathbf{z}_j^T\right]\mathbf{R}^{-1}\right) - 2\boldsymbol{\mu}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{E}\left[\mathbf{z}_j\right] + \boldsymbol{\mu}^T\mathbf{R}^{-1}\boldsymbol{\mu} \qquad (6)$$

In the Expectation step, the posterior means of the spatial topographies are estimated.

$$\bar{\mathbf{z}}_j = \left(\mathbf{x}_j^T\mathbf{x}_j\boldsymbol{\Lambda}^{-1} + \mathbf{R}^{-1}\right)^{-1}(\boldsymbol{\Lambda}^{-1}(\mathbf{Y} - \sum_{i\neq j} \bar{\mathbf{z}}_i\mathbf{x}_i^T)\mathbf{x}_j + \mathbf{R}^{-1}\boldsymbol{\mu}) \qquad (7)$$

For the Maximization step, we use the Conditional Maximization method [32] whereby we sequentially maximize over each one of the parameters $\Theta = \left\{\boldsymbol{\mu}, \mathbf{R}, \boldsymbol{\Lambda}, \tau_j, \right\}$, while holding the others fixed at their previous values.

$$\boldsymbol{\mu} = \frac{1}{J}\sum \bar{\mathbf{z}}_j \qquad (8)$$

$$\mathbf{R} = \frac{1}{JM}\sum\left(\mathbf{S}_j + \bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T - \boldsymbol{\mu}\bar{\mathbf{z}}_j^T - \bar{\mathbf{z}}_j\boldsymbol{\mu}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T\right) \qquad (9)$$

$$\boldsymbol{\Lambda} = \frac{1}{T}\mathbf{Y}\mathbf{Y}^T - \mathbf{Y}\left(\sum \bar{\mathbf{z}}_j\mathbf{x}_j^T\right)^T - \left(\sum \bar{\mathbf{z}}_j\mathbf{x}_j^T\right)\mathbf{Y}^T \\ + \sum_j \left(\mathbf{x}_j^T\mathbf{x}_j\left(\mathbf{S}_j + \bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T\right)^T + \sum_{i\neq j} \mathbf{x}_j^T\mathbf{x}_i\bar{\mathbf{z}}_i\bar{\mathbf{z}}_j^T\right) \qquad (10)$$

The latencies τ_j can be estimated in a similar manner to the single channel SP algorithm using a linear search to maximize $tr\left[\left(\mathbf{Y} - \sum_{i\neq j} \bar{\mathbf{z}}_i\mathbf{x}_i^T\right)^T\boldsymbol{\Lambda}^{-1}\bar{\mathbf{z}}_j\mathbf{x}_j^T\right]$ over the component basis. The complete EM-SP algorithm is provided below.

Algorithm 2: EM-SP

```
Inputs: Multi-channel data $\mathbf{Y} \in \mathbb{R}^{M\times T}$, $\mathbf{X} \in \mathbb{R}^{T\times K}$, the number of components $J$ and latency windows $W_j$
Initialize parameters $\bar{\mathbf{z}}_j$ and $\Theta^0 = \{\tau_j^0, \boldsymbol{\mu}^0, \mathbf{R}^0, \boldsymbol{\Lambda}^0\}$.
repeat for $t = 1, 2, \ldots$
    E-step: Estimate the spatial topographies $\bar{\mathbf{z}}_j$ using (7)
    CM-steps: Estimate parameters $\boldsymbol{\mu}^t, \mathbf{R}^t, \boldsymbol{\Lambda}^t$ using (8)-(10)
    CM-step: Estimate the latencies $\tau_j^t$ using SP as shown below
    Initialize residual $\mathbf{Y}_R^0 = \mathbf{Y}$ and component set $\mathcal{C}^0 = \emptyset$
    Normalize the spatial topographies $\bar{\mathbf{z}}_j = \bar{\mathbf{z}}_j/\max(|\bar{\mathbf{z}}_j|)$
    repeat for iterations $l = 1,2, \ldots$
        repeat for components $j = 1, \ldots, J$
            Find the best component latency
                $\mathbf{c}_j^* = \underset{\tau\in W_j}{\text{argmax}}\, tr\left((\mathbf{Y}_R^{l-1})^T \boldsymbol{\Lambda}^{-1} \bar{\mathbf{z}}_j (\mathbf{X}\mathbf{c}_\tau)^T\right)$
              where $\mathbf{c}_\tau$ is the basis component with latency $\tau$
        Add the $J$ new components to the set $\tilde{\mathcal{C}} = \mathcal{C}^{l-1} \cup \{\mathbf{c}_j^*\}$
        Estimate amplitudes $\tilde{\boldsymbol{\alpha}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$
          where $\mathbf{y} = vec(\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{Y})$ is the vectorized whitened data
          and $\mathbf{A}$ has columns $\left\{vec(\boldsymbol{\Lambda}^{-\frac{1}{2}}\bar{\mathbf{z}}_j(\mathbf{X}\mathbf{c}_j)^T) \;\middle|\; \mathbf{c}_j \in \tilde{\mathcal{C}}\right\}$
        Update $\mathcal{C}^l = \{J$ components with the largest amplitudes
                      for each $W_j\}$
        Re-estimate amplitudes $\boldsymbol{\alpha}^l = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y}$
          where $\mathbf{B}$ has columns $\left\{vec\left(\boldsymbol{\Lambda}^{-\frac{1}{2}}\bar{\mathbf{z}}_j(\mathbf{X}\mathbf{c}_j)^T\right) \middle| \mathbf{c}_j \in \mathcal{C}^l\right\}$
        Calculate the new residual $\mathbf{Y}_R^l = \mathbf{Y} - \sum_j a_j \bar{\mathbf{z}}_j (\mathbf{X}\mathbf{c}_j)^T$
          where $a_j$ are the values in $\boldsymbol{\alpha}^l$
        If $\|\mathbf{Y}_R^l\| > \|\mathbf{Y}_R^{l-1}\|$ stop iterations, let $\mathcal{C}^l = \mathcal{C}^{l-1}$ & $\boldsymbol{\alpha}^l = \boldsymbol{\alpha}^{l-1}$
    Update the spatial topographies $\bar{\mathbf{z}}_j = a_j \bar{\mathbf{z}}_j$
Output: The estimated TRF $\boldsymbol{\beta} = \sum_{j=1}^{J} \bar{\mathbf{z}}_j \mathbf{c}_j^T$, spatial topographies $\bar{\mathbf{z}}_j$, and components $\mathbf{c}_j$ with latencies $\tau_j$ and amplitudes $a_j = \max(|\bar{\mathbf{z}}_j|)$.
```

All four algorithms can also be used to simultaneously fit TRFs to multiple predictors (e.g., foreground and background envelopes) by concatenating the P predictor matrices $X_p \in \mathbb{R}^{T\times K}$ along the columns, resulting in a new predictor matrix $X \in \mathbb{R}^{T\times KP}$. In this work, we jointly fit TRFs to two predictors (corresponding to foreground and background speech envelopes) using a concatenated predictor matrix.

D. Simulation Study

Simulations were constructed to match typical cocktail party speech experiments which have two simultaneous speech streams. Accordingly, the envelopes of two speech stimuli (foreground and background) were used as predictors. These envelopes were constructed by first passing the speech waveform through a gammatone filterbank with 256 frequency bands between 20-4000 Hz, and the amplitude spectrogram was computed with an integration window of 10 ms. The resulting spectrogram was averaged across frequency bands, downsampled to 1000 Hz, and then band-passed at 1-10 Hz using a symmetric linear phase FIR filter with order 3301 and cutoffs 0.5 Hz and 11.25 Hz. Finally, the envelopes were downsampled to 100 Hz for all further analysis. These envelopes were repeated three times, in line with experiments having multiple trials of repeated stimuli to extract consistent responses using spatial filters such as Denoising Source Separation (DSS [33]; details given below). Each predictor was convolved with its own ground truth simulated TRF and the responses were summed together to form one-dimensional responses at 100 Hz sampling rate for 30 pseudo-subjects comparable to a single-sensor M/EEG response or the first auditory response component after DSS.

For each simulated subject, the ground truth simulated TRF was formed by placing Hamming windows of 50 ms width with peaks in the latency ranges 30-80 ms, 90-170 ms and 190-250 ms corresponding to typical latencies of the M50, M100 and M200 components. The M100 component was given a negative sign, and the components were scaled and shifted according to randomized subject specific amplitudes and latencies. These amplitudes and latencies were later used as the ground truth for performance evaluation.

Realistic noise was added to the simulated responses using the first DSS component of real MEG data collected from 30 subjects listening to speech (previously published [34], [35]). DSS creates a series of spatial filters, where the DSS component generated by the first of these filters corresponds to activity that is most consistent across repeated stimulus presentations (see [33] for details on DSS). Therefore, for this speech experiment, the first DSS component is dominated by auditory activity and displays a typical auditory response sensor topography. This component was then phase scrambled, preserving the spectral properties of MEG signals, to simulate noise added to the simulated response, at SNRs of -15, -20, -25 and -30 dB (SNRs selected to result in realistic TRF model correlation values).

The multi-channel simulation followed the same method for 157 simulated sensor signals, but in addition also used ground truth sensor topographies for each TRF component. These topographies were constructed from the TRF component topographies of a real subject with typical auditory TRF components, with the addition of Gaussian noise to simulate individual variability. Real multi-channel MEG data was again phase scrambled and added as noise on a per channel basis using the method described above, at SNRs of -20, -25, -30 and -35 dB (lower SNRs were used because unprocessed multi-channel data is typically noisier than the extracted auditory component).

The DSS algorithm was also applied to the simulated multi-channel data and corresponding TRFs were calculated for the first 6 DSS components. These DSS TRFs were projected back into sensor space for subsequent analysis and for computing performance metrics.

The source space simulation was constructed using the Freesurfer ico-4 surface source space of the ‘fsaverage’ brain [36]. An ROI in temporal lobe with 245 sources that included auditory cortex was used for this simulation (‘aparc’ labels ‘transversetemporal’ and ‘superiortemporal’). The three TRF components were simulated using dipoles in Heschl’s gyrus, Planum Temporale and Superior Temporal Gyrus in both hemispheres. These dipoles were projected onto the sensors using forward models from real data and back projected back onto source space with Minimum Norm Estimation (MNE) [37] using Eelbrain [14], [38] and MNE-Python softwares [39] to simulate the source localization procedure. The back-projected source distributions of these simulated TRF components were also used as the ground truth for subsequent performance comparisons. The TRFs were then convolved with the predictors to form the responses at each of the 245 sources. Real MEG data was phase scrambled and added as noise to the response at each source at SNRs of -15, -20, -25 and -30 dB following the same procedure as above.

E. Experimental Dataset

MEG data collected in a prior study [34], [35] was used for evaluating the performance of the algorithms on real data. The study was approved by the IRB of the University of Maryland and all participants provided written informed consent prior to the start of the experiment. The dataset consisted of MEG data collected from 40 subjects while they listened to speech from the narration of an audiobook. Subjects listened to two speakers simultaneously in a cocktail party experiment, but were asked to attend to only one speaker. The data was from the condition where the foreground speaker was 3 dB louder than the background speaker. TRFs were estimated for the foreground and background envelopes. Whole head sensor space TRFs (157 sensors) were computed for each algorithm on three minutes of data. Additionally, TRFs were also computed for the first 6 DSS components. Finally, the MEG responses of this dataset were source localized using MNE and source space TRFs were also computed.

F. Algorithm Implementation

The algorithms were implemented in Python (version 3.9.6) using SciPy (version 1.8.0) [40], and Eelbrain (version 0.36.1). The code is available online at <URLs available upon acceptance>. A basis dictionary with Hamming windows of width 50 ms was used for boosting, SP and EM-SP. The component latency windows for the SP and EM-SP algorithms were 30-80 ms, 90-170 ms and 190-250 ms. To avoid instability and convergence issues, the spatial covariance $\mathbf{R}$ for the EM-SP algorithm was assumed to be the identity matrix. The EM-SP was initialized using the extracted components from the SP algorithm applied at each sensor/source independently.

A nested 4-fold cross validation procedure was followed for all algorithms to allow for unbiased comparison. The data was divided into 4 splits, with 1 for testing, 1 for validation and 2 for training. The validation and training splits were permuted for each test split in a nested fashion. The training data was used to optimize the ridge TRF over several regularization parameters (steps of 2^0, 2^1, ..., 2^{16}) based on the model fit on the validation data. The boosting TRF was fit on the training data, and the validation data was used to check for convergence and terminate the algorithm. The SP and EM-SP TRFs were fit on the training data, and the model fit on the validation data was used to terminate the EM iterations. Finally, the overall model fit metric was calculated by convolving the average TRF over all training splits with the appropriate test predictor and computing the Pearson correlation between this predicted signal and the actual test signal.

G. Performance Metrics

The model fit was calculated as the Pearson correlation between the estimated and the predicted response (averaged over channels for multidimensional cases). A null model was constructed by fitting TRFs using circularly time-shifted predictors (shifts of 15 s) and the correlation of this null model was subtracted from the true model. This bias corrected model fit is reported for both simulations and real data.

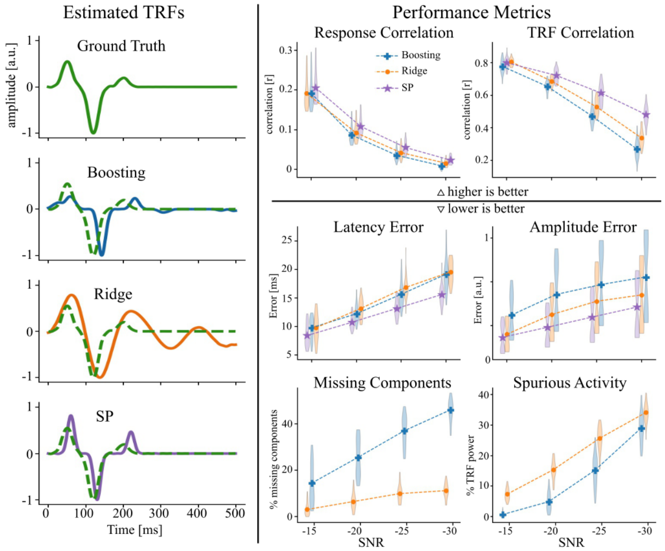
In addition to model fit, several other metrics of TRF component estimation were also calculated for the simulations (but not for real data, since the ground truth components were unknown). TRF components were automatically detected as the peaks of the r.m.s of the TRF across channels in the appropriate latency windows (30-80 ms, 90-170 ms, 190-250 ms) and the following metrics were used; 1) Pearson correlation between the estimated and ground truth TRF, 2) Absolute error of individual component latency estimates 3) Absolute error of individual component amplitude estimates (estimated vs, ground truth), 4) Spurious TRF activity given by the % r.m.s. power in the estimated TRF after 300 ms (note that there is no activity in the ground truth TRF after 300 ms), 5) Number of missing components 6) Sensor/source topography estimation error using the angle between the estimated topography vector and the ground truth topography vector. These metrics were averaged over channels, predictors, and components.

III. Results

A. *Simulation: Single-Channel TRFs*

Single-channel TRFs were simulated, and the ridge, boosting, and SP algorithms were compared in terms of several performance metrics. The estimated TRFs for a representative subject are shown in Fig. 1. The conventional measure for evaluating the performance of TRF models is the correlation between the actual and the predicted responses. In this work we used a nested cross-validation procedure for all algorithms to reduce overfitting and a null model based on shifted predictors for bias correction. However, correlation between the actual and the predicted responses may not always be an appropriate measure of TRF component estimation, since it depends on a variety of factors including SNR and predictor characteristics. This metric may also not appropriately penalize latency errors or spurious activity in the TRF. Hence, we used several other metrics, including component latency and amplitude errors, to compare these algorithms in terms of TRF component estimation (see right column of Fig. 1).

The SP algorithm performed the best in most measures, while ridge and boosting performed comparably. Spurious peaks after 300 ms (when there was no activity in the ground truth TRF) could lead to difficulties in interpretation and to false positives when detecting TRF components in real data. Conversely, missing components (false negatives) could also lead to improper interpretation of TRFs. Ridge had more spurious activity than boosting but was also able to detect more components than boosting.



(a) (b)

Fig. 1. Performance comparison for single-channel simulations. (a) The fitted TRFs for a representative subject. The ground truth TRF is shown as a dotted green line over the estimated TRFs. (b) Algorithm comparison using the performance metrics. Violin plots over simulated subjects are shown, with the symbols indicating the mean. Within each SNR condition, the algorithms are plotted in ascending order of their means from left to right. SP does not have spurious activity after 300 ms or missing components by design and is not shown for the bottom two subplots. Boosting seems to miss some components, while ridge has more spurious activity. Ridge and boosting are comparable for most measures, while SP seems to outperform the others in higher SNR cases.

B. *Simulation: Multi-channel TRFs*

Sensor space TRFs were simulated using realistic sensor topographies for TRF components, and the performance of each algorithm was compared (see Fig. 2). TRFs were estimated independently at each sensor for the boosting, ridge and SP algorithms, while the EM-SP algorithm directly estimated multi-channel component topographies. The EM-SP algorithm performed the best in most measures, while ridge and boosting performed comparably. The sensor topographies estimated by boosting and SP are worse than those estimated by ridge and EM-SP, which is to be expected given that the former are sparse algorithms that are fit at each sensor independently. Interestingly, the missing components are similar for both ridge and boosting, unlike in the single-channel case. If boosting is able to correctly estimate components even for only a few channels, sparsity (in time) can then preserve the presence of the component peak when the r.m.s of the TRF is taken across channels. This improvement in component detection for boosting is also seen for the DSS and source space TRFs reported below.

C. *Simulation: Denoised TRFs using DSS*

The DSS algorithm was applied to the simulated sensor space responses to extract spatial filters corresponding to auditory response components. The algorithms were fit on the first 6 DSS components, and the resulting TRFs were projected back onto the sensor space for performance evaluation. Model fit response correlations increased greatly over the sensor space TRFs in all cases (see Fig. 3). Ridge, boosting and EM-SP had comparable results. Interestingly, EM-SP did not have a significant advantage over the other algorithms, indicating that the established algorithms are just as suitable for low dimensional, denoised data.

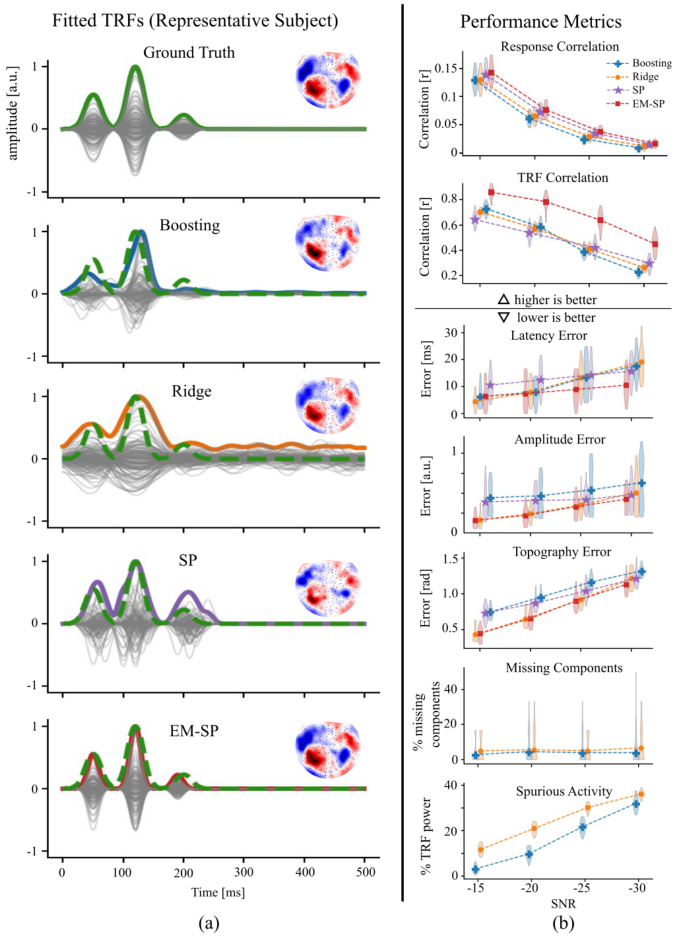


Fig. 2. Performance comparison for multi-channel simulations. (a) The fitted TRFs for a representative subject. The TRF at each sensor is plotted in gray, while the ℓ_2-norm over sensors is plotted as a colored thick line. The ℓ_2-norm of the ground truth TRF is shown as a dotted green line over the estimated TRFs. The sensor topography at the largest peak near 100 ms is shown as an inset. (b) Algorithm comparison using the performance metrics. Since there is no activity after 300 ms in the SP and EM-SP TRFs by design, they are not plotted in the spurious activity subplot. EM-SP outperforms the others in most measures. Although all methods find similar components, the sensor topographies for boosting and SP are worse than the others, perhaps because they are sparse estimation techniques.

D. *Simulation: Source Localized TRFs*

Source space simulations were constructed with dipoles in auditory areas for each TRF component. These dipoles were projected onto sensor space using the forward model and source localized back to source space to simulate source localized MEG data. The algorithms were fit on these source localized signals and performance was compared using the same metrics (see Fig. 4). Results were similar to the sensor space simulation, with EM-SP outperforming the others, and ridge and boosting giving comparable results (with ridge typically performing marginally better than boosting for most measures except spurious activity).

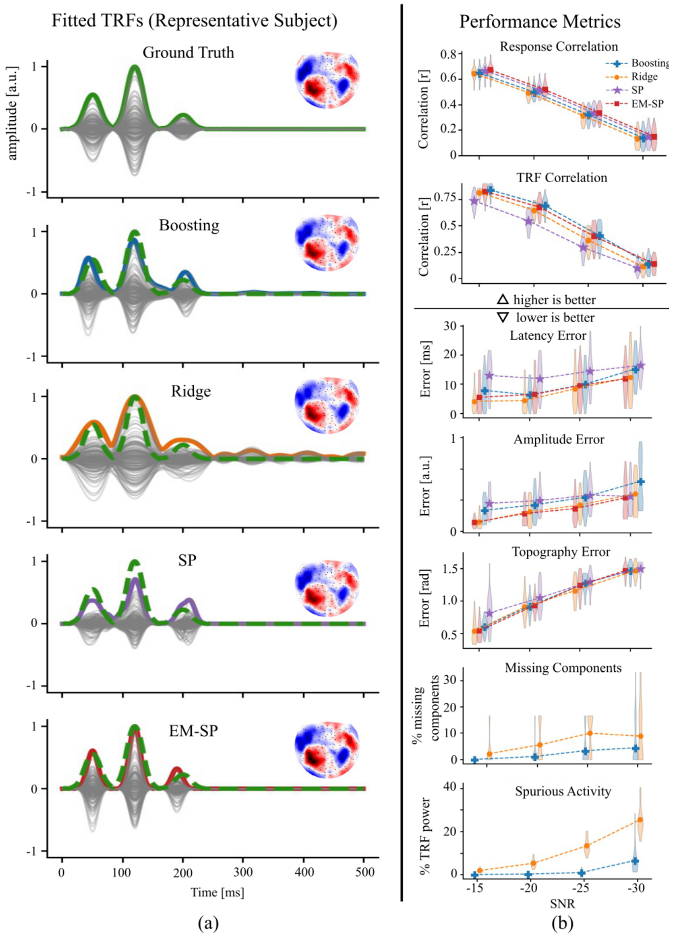


Fig. 3. Performance comparison after DSS denoising. (a). The fitted TRFs for a representative subject, similar to the previous figure. The TRFs were fit on the first 6 DSS components and then back-projected to sensor space. All the algorithms result in reasonable TRF components and sensor topographies. (b). Algorithm comparison using the performance metrics. All the algorithms except SP perform comparably, while the latter performs the worst in most cases.

Overall, the simulation results indicate that both boosting and ridge are comparable, with ridge typically performing slightly better. Interestingly, SP outperformed ridge and boosting in the high noise single-channel simulations, while EM-SP outperformed the others by a large margin in the multi-channel and source-localized simulations. It should be noted that the component windows used for the simulation were identical to the component windows provided a-priori to SP and EM-SP, which may explain their better performance. Therefore, SP and EM-SP may be suitable for estimating TRFs in high noise conditions, assuming that the appropriate latency windows can be determined a-priori. Ridge also had lower spatial error compared to boosting (sensor topography and source distribution errors), perhaps because a sparse estimation technique like boosting cannot capture smooth spatial patterns as well as ridge. Conversely, ridge had much larger amounts of spurious activity compared to boosting. However, after applying the DSS algorithm, ridge, boosting and EM-SP once again showed comparable performance, highlighting the importance of denoising methods when estimating TRFs from noisy multidimensional data.

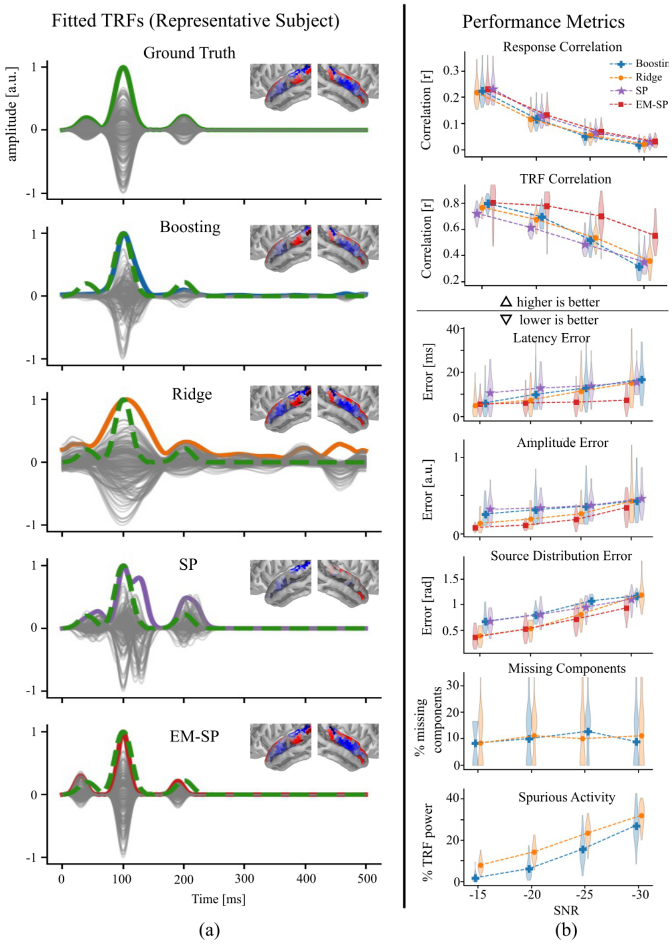
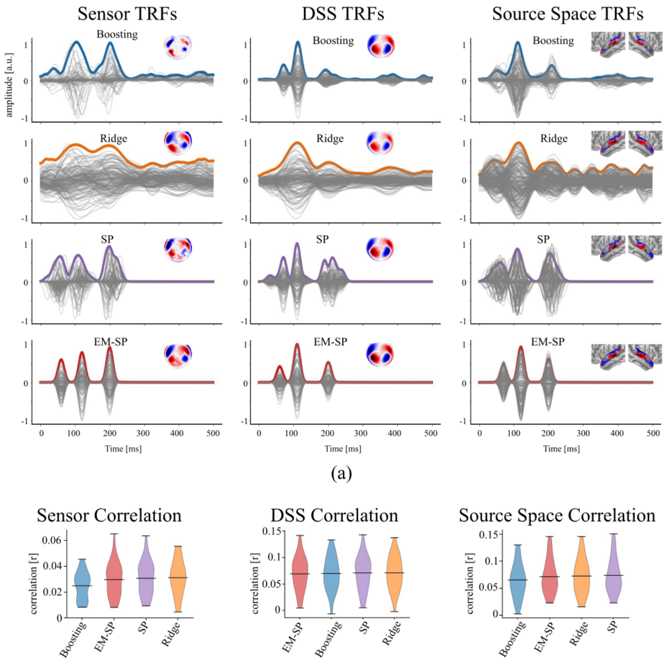


Fig. 4. Performance comparison for source space simulations. (a) The fitted TRFs for a representative subject are shown, similar to the previous figure. The source distributions in the temporal lobe ROI at the largest peak near 100 ms are shown as insets. Boosting and SP result in much sparser source distributions, and all the algorithms except SP perform comparably in estimating the TRF components, although the ridge TRF has a lot more activity that may make it difficult to interpret in realistic situations where the ground truth is unknown. (b). Algorithm comparison using the performance metrics, similar to those shown in the previous figure. EM-SP outperforms the others in most cases.

E. Performance on Real Data

The algorithms were compared on a real MEG dataset collected for a cocktail party experiment. Sensor space, DSS and source space TRFs are shown for a representative subject in Fig. 5. The only metric used was the correlation between the measured and predicted signals, since the other metrics cannot be calculated when the ground truth TRF components are unknown. Interestingly, boosting had significantly lower correlation accuracy compared to each of the three other algorithms for sensor and source space TRFs (paired samples permutation tests with Holm-Bonferroni correction; all comparisons with boosting resulted in $t_{39} > 4$, $p < 0.01$), but there were no significant differences in correlation accuracy between ridge, SP and EM-SP. However, it is unclear if correlation is the most suitable metric for evaluating the accuracy of estimating TRF components. The correlation values were distributed over a large range across subjects, possibly indicating a high degree of inter-subject variability in neural SNR for time-locked responses. Ridge resulted in smooth TRFs with several peaks and large amounts of non-zero activity which made them more difficult to interpret, especially for the sensor and source space TRFs. Boosting, though performing worse in terms of correlation, allowed for sparser TRFs with fewer peaks that were easier to interpret.



(a)

(b)

Fig. 5. Performance comparison on real MEG data. (a) The estimated sensor, DSS and source localized TRFs are shown for a representative subject. The sensor topographies and source distributions at the large peak near 100 ms are shown as insets. The sensor space EM-SP TRF has clear components and topographies, while the boosting TRF has overly sparse topographies and the ridge TRF has a lot of hard to interpret activity. Boosting, ridge and EM-SP show clear components and spatial patterns for the DSS and source localized TRFs. (b) Correlation between the measured and predicted signals is shown as a measure of model fit. Violin plots across subjects are shown for each algorithm in ascending order of their mean from left to right.

The two proposed algorithms were restricted to finding exactly three TRF components, assuming fixed component waveforms and latency windows. The fact that EM-SP may have performed worse than ridge for real data, even though it outperformed the others in the simulations, indicates that these assumptions may not be valid for all subjects. This could be due to a variety of reasons including missing components due to anatomical or functional differences, and large individual variability in TRF components latencies, waveforms, and peak widths. Indeed, a separate simulation analysis (not shown) with missing components and mismatched latency windows resulted in similar performance for EM-SP, with it no longer outperforming ridge and boosting. In any case, conventional post-hoc analysis of TRF components estimated using established algorithms is also typically performed under similar assumptions to those used for EM-SP (i.e., detecting TRF peaks using similar latency windows). However, even with these constraints, EM-SP was often able to recover TRF components and spatial patterns comparable to ridge.

IV. Conclusion

TRFs provide a significant advancement over ERPs, allowing for experiments with more naturalistic speech paradigms. Detecting robust TRF components is essential for reliable single-subject investigations that could inform diagnosis and treatment of hearing disabilities and lead to improved biomedical applications like smart hearing aids.

We compared TRF algorithms using both model fit and component estimation accuracy. Simulations indicate that boosting and ridge are comparable for most cases. Interestingly, ridge had better model fits on real data. However, in general, ridge TRFs displayed more spurious activity, while boosting TRF peaks were more interpretable. Therefore, ridge may be suitable for studies focused on prediction accuracy, while boosting may be appropriate for detecting easily identifiable TRF components. We restricted our analysis of established methods to these two algorithms that are the most widely used. Other variations on regularized regression, such as Lasso and Elastic Net, may provide improvements in TRF estimation [12].

SP and EM-SP performed exceptionally in simulations, but not on real data, possibly due to invalid assumptions. The a-priori parameters may need to be tuned for each predictor type or experiment, or even for each subject

Modern TRF analyses involve multiple types of predictors [42] (e.g., envelopes, phoneme onsets, multiple frequency bands for spectrotemporal TRFs). Boosting and banded ridge regression may be suitable for these studies [10], [13], [43], [44]. The component characteristics of TRFs to these higher-level predictors must be determined before SP and EM-SP can be applied. Additionally, early low-level responses could impact TRFs to high-level predictors, and sparse algorithms with fewer false positives (but more false negatives) may be more conservative.

In conclusion, our results indicate that SP and EM-SP may only perform well under realistic assumptions, while ridge and boosting perform comparably in most cases, with ridge typically having higher prediction accuracies, but also more spurious activity.

References

[1] T. Picton, “Hearing in Time: Evoked Potential Studies of Temporal Processing,” *Ear and Hearing*, vol. 34, no. 4, pp. 385–401, 2013, doi: 10.1097/AUD.0b013e31827ada02.

[2] T. W. Picton, S. A. Hillyard, H. I. Krausz, and R. Galambos, “Human auditory evoked potentials. I: Evaluation of components,” *Electroencephalography and Clinical Neurophysiology*, vol. 36, pp. 179–190, Jan. 1974, doi: 10.1016/0013-4694(74)90155-2.

[3] N. Ding and J. Z. Simon, “Emergence of neural encoding of auditory objects while listening to competing speakers,” *PNAS*, vol. 109, no. 29, pp. 11854–11859, Jul. 2012, doi: 10.1073/pnas.1205381109.

[4] E. C. Lalor and J. J. Foxe, “Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution,” *European Journal of Neuroscience*, vol. 31, no. 1, pp. 189–193, 2010, doi: 10.1111/j.1460-9568.2009.07055.x.

[5] 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, Aug. 2017, doi: 10.1109/TBME.2016.2628884.

[6] S. Geirnaert *et al.*, “Electroencephalography-Based Auditory Attention Decoding: Toward Neurosteered Hearing Devices,” *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 89–102, Jul. 2021, doi: 10.1109/MSP.2021.3075932.

[7] C. Brodbeck, A. Jiao, L. E. Hong, and J. Z. Simon, “Neural speech restoration at the cocktail party: Auditory cortex recovers masked speech of both attended and ignored speakers,” *PLOS Biology*, vol. 18, no. 10, p. e3000883, Oct. 2020, doi: 10.1371/journal.pbio.3000883.

[8] 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, May 2018, doi: 10.1016/j.neuroimage.2018.01.042.

[9] 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.e3, Mar. 2018, doi: 10.1016/j.cub.2018.01.080.

[10] C. Brodbeck, L. E. Hong, and J. Z. Simon, “Rapid Transformation from Auditory to Linguistic Representations of Continuous Speech,” *Current Biology*, vol. 28, no. 24, pp. 3976-3983.e5, Dec. 2018, doi: 10.1016/j.cub.2018.10.042.

[11] C. Brodbeck, A. Presacco, S. Anderson, and J. Z. Simon, “Over-Representation of Speech in Older Adults Originates from Early Response in Higher Order Auditory Cortex,” *Acta Acustica united with Acustica*, vol. 104, no. 5, pp. 774–777, Sep. 2018, doi: 10.3813/AAA.919221.

[12] D. D. E. Wong, S. A. Fuglsang, J. Hjortkjær, E. Ceolini, M. Slaney, and A. de Cheveigné, “A Comparison of Regularization Methods in Forward and Backward Models for Auditory Attention Decoding,” *Front. Neurosci.*, vol. 12, 2018, doi: 10.3389/fnins.2018.00531.

[13] M. J. Crosse, N. J. Zuk, G. M. Di Liberto, A. R. Nidiffer, S. Molholm, and E. C. Lalor, “Linear Modeling of Neurophysiological Responses to Speech and Other Continuous Stimuli: Methodological Considerations for Applied Research,” *Front Neurosci*, vol. 15, p. 705621, Nov. 2021, doi: 10.3389/fnins.2021.705621.

[14] C. Brodbeck *et al.*, “Eelbrain: A Python toolkit for time-continuous analysis with temporal response functions,” Aug. 2021. doi: 10.1101/2021.08.01.454687.

[15] S. V. David, N. Mesgarani, and S. A. Shamma, “Estimating sparse spectro-temporal receptive fields with natural stimuli,” *Network*, vol. 18, no. 3, pp. 191–212, Sep. 2007, doi: 10.1080/09548980701609235.

[16] M. J. Crosse, G. M. Di Liberto, A. Bednar, and E. C. Lalor, “The Multivariate Temporal Response Function (mTRF) Toolbox: A MATLAB Toolbox for Relating Neural Signals to Continuous Stimuli,” *Front. Hum. Neurosci.*, vol. 0, 2016, doi: 10.3389/fnhum.2016.00604.

[17] C. D. Woody, "Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals," *Medical & Biological Engineering*, vol. 5, no. 6, pp. 539–554, Nov. 1967, doi: 10.1007/BF02474247.

[18] T.-P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski, "Analyzing and Visualizing Single-Trial Event-Related Potentials," in *Advances in Neural Information Processing Systems 11*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, 1999, pp. 118–124. Accessed: Oct. 02, 2020. [Online]. Available: http://papers.nips.cc/paper/1574-analyzing-and-visualizing-single-trial-event-related-potentials.pdf

[19] S. Makeig *et al.*, "Dynamic Brain Sources of Visual Evoked Responses," *Science*, vol. 295, no. 5555, pp. 690–694, Jan. 2002, doi: 10.1126/science.1066168.

[20] R. Q. Quiroga and H. Garcia, "Single-trial event-related potentials with wavelet denoising," *Clinical Neurophysiology*, vol. 114, no. 2, pp. 376–390, Feb. 2003, doi: 10.1016/S1388-2457(02)00365-6.

[21] J. C. de Munck, F. Bijma, P. Gaura, C. A. Sieluzycki, M. I. Branco, and R. M. Heethaar, "A maximum-likelihood estimator for trial-to-trial variations in noisy MEG/EEG data sets," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 12, pp. 2123–2128, Dec. 2004, doi: 10.1109/TBME.2004.836515.

[22] P. Jaskowski and R. Verleger, "Amplitudes and latencies of single-trial ERP's estimated by a maximum-likelihood method," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 8, pp. 987–993, Aug. 1999, doi: 10.1109/10.775409.

[23] L. Xu, P. Stoica, J. Li, S. L. Bressler, X. Shao, and M. Ding, "ASEO: A Method for the Simultaneous Estimation of Single-Trial Event-Related Potentials and Ongoing Brain Activities," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 1, pp. 111–121, Jan. 2009, doi: 10.1109/TBME.2008.2008166.

[24] T. Limpiti, B. D. Van Veen, and R. T. Wakai, "A Spatiotemporal Framework for MEG/EEG Evoked Response Amplitude and Latency Variability Estimation," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 616–625, Mar. 2010, doi: 10.1109/TBME.2009.2032533.

[25] C. Sieluzycki, R. Konig, A. Matysiak, R. Kus, D. Ircha, and P. J. Durka, "Single-Trial Evoked Brain Responses Modeled by Multivariate Matching Pursuit," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 1, pp. 74–82, Jan. 2009, doi: 10.1109/TBME.2008.2002151.

[26] H. R. Mohseni, F. Ghaderi, E. L. Wilding, and S. Sanei, "Variational Bayes for Spatiotemporal Identification of Event-Related Potential Subcomponents," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2413–2428, Oct. 2010, doi: 10.1109/TBME.2010.2050318.

[27] W. Wu, C. Wu, S. Gao, B. Liu, Y. Li, and X. Gao, "Bayesian estimation of ERP components from multicondition and multichannel EEG," *NeuroImage*, vol. 88, pp. 319–339, Mar. 2014, doi: 10.1016/j.neuroimage.2013.11.028.

[28] W. Dai and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009, doi: 10.1109/TIT.2009.2016006.

[29] 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)*, vol. 39, no. 1, pp. 1–22, 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

[30] C. B. Do and S. Batzoglou, "What is the expectation maximization algorithm?," *Nat Biotechnol*, vol. 26, no. 8, pp. 897–899, Aug. 2008, doi: 10.1038/nbt1406.

[31] T. Zhang and B. Yu, "Boosting with early stopping: Convergence and consistency," *The Annals of Statistics*, vol. 33, no. 4, pp. 1538–1579, Aug. 2005, doi: 10.1214/009053605000000255.

[32] X.-L. Meng and D. B. Rubin, "Maximum Likelihood Estimation via the ECM Algorithm: A General Framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993, doi: 10.2307/2337198.

[33] A. de Cheveigné and J. Z. Simon, "Denoising based on spatial filtering," *J Neurosci Methods*, vol. 171, no. 2, pp. 331–339, Jun. 2008, doi: 10.1016/j.jneumeth.2008.03.015.

[34] A. Presacco, J. Z. Simon, and S. Anderson, "Evidence of degraded representation of speech in noise, in the aging midbrain and cortex," *J Neurophysiol*, vol. 116, no. 5, pp. 2346–2355, Nov. 2016, doi: 10.1152/jn.00372.2016.

[35] A. Presacco, J. Z. Simon, and S. Anderson, "Effect of informational content of noise on speech representation in the aging midbrain and cortex," *Journal of Neurophysiology*, vol. 116, no. 5, pp. 2356–2367, Nov. 2016, doi: 10.1152/jn.00373.2016.

[36] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, Aug. 2012, doi: 10.1016/j.neuroimage.2012.01.021.

[37] M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting magnetic fields of the brain: minimum norm estimates," *Med. Biol. Eng. Comput.*, vol. 32, no. 1, pp. 35–42, Jan. 1994, doi: 10.1007/BF02512476.

[38] C. Brodbeck, P. Das, J. P. Kulasingham, S. Reddigari, and T. L. Brooks, *Eelbrain 0.36*. Zenodo, 2021. doi: 10.5281/zenodo.5152554.

[39] A. Gramfort *et al.*, "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446–460, Feb. 2014, doi: 10.1016/j.neuroimage.2013.10.027.

[40] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, Art. no. 3, Mar. 2020, doi: 10.1038/s41592-019-0686-2.

[41] C. Brodbeck and J. Z. Simon, "Continuous speech processing," *Current Opinion in Physiology*, vol. 18, pp. 25–31, Dec. 2020, doi: 10.1016/j.cophys.2020.07.014.

[42] M. Gillis, J. Vanthornhout, J. Z. Simon, T. Francart, and C. Brodbeck, "Neural Markers of Speech Comprehension: Measuring EEG Tracking of Linguistic Speech Representations, Controlling the Speech Acoustics," *J. Neurosci.*, vol. 41, no. 50, pp. 10316–10329, Dec. 2021, doi: 10.1523/JNEUROSCI.0812-21.2021.

[43] J. P. Kulasingham, N. H. Joshi, M. Rezaeizadeh, and J. Z. Simon, "Cortical Processing of Arithmetic and Simple Sentences in an Auditory Attention Task," *J. Neurosci.*, vol. 41, no. 38, pp. 8023–8039, Sep. 2021, doi: 10.1523/JNEUROSCI.0269-21.2021.

[44] J. P. Kulasingham, C. Brodbeck, A. Presacco, S. E. Kuchinsky, S. Anderson, and J. Z. Simon, "High gamma cortical processing of continuous speech in younger and older listeners," *NeuroImage*, vol. 222, p. 117291, Nov. 2020, doi: 10.1016/j.neuroimage.2020.117291.