

# Accurate Modeling of Brain Responses to Speech

Daniel D.E. Wong<sup>\*1,2</sup>, Giovanni M. Di Liberto<sup>†1,2</sup>, and Alain de Cheveigné<sup>‡1,2,3</sup>

<sup>1</sup>*Laboratoire des Systèmes Perceptifs, UMR 8248, CNRS, Paris, France*

<sup>2</sup>*Département d'Études Cognitives, École Normale Supérieure, Université PSL, Paris, France*

<sup>3</sup>*Ear Institute, University College London, London, United Kingdom*

December 31, 2018

## Abstract

1  
2 Perceptual processes can be probed by fitting stimulus-response models  
3 that relate measured brain signals such as electroencephalography (EEG) to  
4 the stimuli that evoke them. These models have also found application for  
5 the control of devices such as hearing aids. The quality of the fit, as measured  
6 by correlation, classification, or information rate metrics, indicates the value  
7 of the model and the usefulness of the device. Models based on Canonical  
8 Correlation Analysis (CCA) achieve a quality of fit that surpasses that of  
9 commonly-used linear forward and backward models. Here, we show that  
10 their performance can be further improved using several techniques, includ-  
11 ing adaptive beamforming, CCA weight optimization, and recurrent neural  
12 networks that capture the time-varying and context-dependent relationships  
13 within the data. We demonstrate these results using a match-vs-mismatch  
14 classification paradigm, in which the classifier must decide which of two stim-  
15 ulus samples produced a given EEG response and which is a randomly chosen  
16 stimulus sample. This task captures the essential features of the more com-  
17 plex auditory attention decoding (AAD) task explored in many other studies.  
18 The new techniques yield a significant decrease in classification errors and an  
19 increase in information transfer rate, suggesting that these models better fit  
20 the perceptual processes reflected by the data. This is useful for improving  
21 brain-computer interface (BCI) applications.

---

\*daniel.wong@ens.fr

†diliberg@tcd.ie

‡alain.de.cheveigne@ens.fr

## 22 1 Introduction

23 In experiments that record brain responses to stimulation, stimulus-response models  
24 are useful in providing insight into the cortical components of the response. As these  
25 models can provide information about auditory attention, they have also been put  
26 forward for brain-computer interface (BCI) applications, such as the “cognitive”  
27 control of a hearing aid [Wronkiewicz et al., 2016]. Previous studies have used linear  
28 system identification techniques to either predict the response from the stimulus  
29 (forward model) or else infer the stimulus from the response (backward model)  
30 [Lalor and Foxe, 2010, Ding and Simon, 2012a,b, 2013, 2014]. In addition to these, a  
31 third form of model projects both stimulus and response into a common subspace via  
32 weight matrices obtained using Canonical Correlation Analysis (CCA) [Hotelling,  
33 1936, Dmochowski et al., 2017, de Cheveigné et al., 2018]. As they are applicable to  
34 responses to arbitrary stimuli, they allow the research to move beyond the standard  
35 “evoked-response” paradigm that requires repeating the same short stimulus many  
36 times [Ross et al., 2010]. The quality of the model can be quantified by calculating  
37 the correlation coefficient between actual and predicted brain response (forward  
38 model), or between the actual and inferred stimulus (backward model), or between  
39 canonical correlate (CC) pairs (CCA). Higher correlation values indicate that the  
40 model better captures the relation between stimulus and response.

41 Alternatively, the quality of a model can be quantified on the basis of its per-  
42 formance in a classification task, in terms of discriminability ( $d$ -prime) or percent  
43 correct classification. This is particularly useful when developing a model for BCI  
44 applications where classification decisions are made based on short segments of  
45 data. In this paper, we use a simple “match-vs-mismatch” task based on the cor-  
46 tical response to a single speech stream [de Cheveigné et al., 2018], in which the  
47 classifier must decide whether a segment of EEG matches the segment of stimulus  
48 that evoked it, as opposed to some unrelated segment of the same stimulus. A good  
49 classification performance is taken to indicate that the model successfully captures  
50 the stimulus-response relationship.

51 Other studies have used the more complex Auditory Attention Decoding (AAD)  
52 task, in which a subject is presented with two concurrent stimulus streams (for  
53 example two voices speaking at the same time) and required to attend one stream  
54 or the other. The classifier attempts to identify which stream was the focus of  
55 the subject’s attention, given both stimulus streams and the EEG [Hillyard et al.,  
56 1973, Ding and Simon, 2012b, Mirkovic et al., 2015, 2016, O’Sullivan et al., 2015,  
57 Akram et al., 2016, O’Sullivan et al., 2017]. Our simpler task allows a more direct  
58 evaluation of the stimulus-response model that underlies both tasks.

59 A previous study from our group found that models based on CCA were superior  
60 to classic forward and backward models in terms of correlation,  $d$ -prime, and classi-

61 fication error rate [de Cheveigné et al., 2018]. Better performance was attributed to  
62 the ability of CCA to strip both stimulus and EEG of irrelevant dimensions, and to  
63 the fact that the multiple CCs allow multivariate classifiers to be deployed. In the  
64 aforementioned study, the various models were constrained to have the same num-  
65 ber of free parameters so as to ensure a fair comparison between models. Here, we  
66 relax that constraint and introduce several new schemes to improve model quality.  
67 Arguably, models that give better performance more accurately capture the cortical  
68 representation of the stimulus, and good performance is also essential for applica-  
69 tions. Each strategy is evaluated individually and in combination with others by  
70 comparison with a baseline (backward model or CCA).

71 Apart from the standard backward model, we test the following models and  
72 classification schemes (each coded by a letter): *CCA* (C), *maximizing component d-*  
73 *prime* (D), *adaptive beamforming* (B), *linear discriminant analysis* (L), *multilayer*  
74 *perceptron* (M), *simple recurrent layer* (S) and *gated recurrent unit* (G). Both *D*  
75 and *B* improve the computation of CCA components. *D* does this during training,  
76 and *B* does this during testing. *M*, *S* and *G* use a neural network architecture to  
77 improve the match-vs-mismatch classification over *L*.

## 78 **2 Methods**

### 79 **2.1 Evaluation Dataset**

80 The dataset used to evaluate canonical correlation analysis (CCA) performance  
81 was presented in [de Cheveigné et al., 2018] and published in [Broderick et al.,  
82 2018a,b]. The speech stimulus was an audio book recording of the “Old Man and  
83 the Sea” recorded with a 44100 Hz sampling rate. The recording was divided  
84 into 32 segments lasting approximately 155s each. The stimulus was presented  
85 diotically over headphones to 8 subjects, while electroencephalography (EEG) data  
86 were recorded using a 128-channel Biosemi system with a sampling rate of 512 Hz.  
87 The subjects heard a single speech stream, in contrast to other studies in which  
88 subjects were presented with two (or more) concurrent speech streams.

### 89 **2.2 Classification Task**

90 Stimulus-response models were evaluated using a classificaton task that involved  
91 deciding which of two candidate speech stream segments gave rise to a given EEG  
92 segment (match-vs-mismatch single-talker classification task). We chose this task,  
93 based on single-talker data, as it permits the analysis to focus on improving the  
94 stimulus-response models and decoding algorithms from a signal processing perspec-  
95 tive rather than dealing with the cortical dynamics of attention that is encountered  
96 in the commonly used AAD task.

## 97 **2.3 EEG and audio preprocessing**

98 We employed the same preprocessing procedures as in [Wong et al., 2018]. In  
99 short, 50 Hz line noise and harmonics were filtered from the EEG using a boxcar  
100 (smoothing) filter kernel of duration 1/50 Hz. The data were then downsampled  
101 to 64 Hz using a resampling method based on the Fast Fourier Transform (FFT).  
102 To downsample, this method reduces the size of the FFT of the signal by truncat-  
103 ing frequency components above the Nyquist frequency. An inverse FFT is then  
104 used to restore the signal to the time domain. The mean was removed from each  
105 EEG channel. EEG was then highpassed at 0.1 Hz using a 4th order forward-pass  
106 Butterworth filter for low frequency detrending. The joint diagonalization frame-  
107 work [de Cheveigné and Parra, 2014] was employed to remove eye artifacts in an  
108 automated fashion as described in [Wong et al., 2018], using FP3 and FP4 chan-  
109 nels to detect eyeblink timepoints. For the the backward model, the EEG data  
110 was further bandpassed between 1-9 Hz using a windowed sinc type I linear-phase  
111 finite-impulse-response (FIR) filter, shifted by its group delay to produce a zero-  
112 phase [Widmann et al., 2015], with a conservatively chosen order of 128 to minimize  
113 ringing effects. This frequency range was chosen as it has been shown that the cor-  
114 tical responses time-lock to speech envelopes in this range [O’Sullivan et al., 2015].

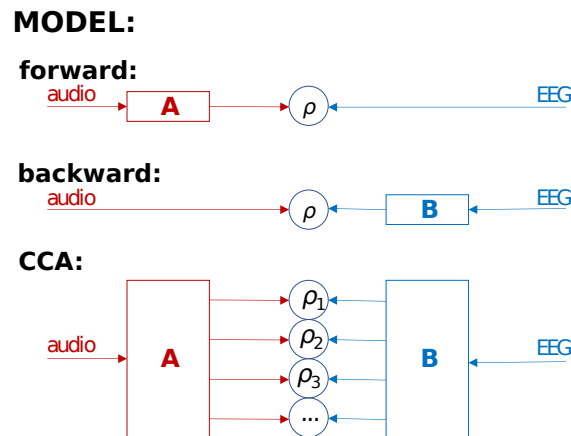
115 To obtain broadband audio envelopes, the presented speech stimuli were filtered  
116 into 31 frequency bands via a gammatone filterbank with a frequency range of 80-  
117 8000Hz [Patterson et al., 1987]. Each frequency band was fullwave rectified and  
118 raised to the power of 0.3 before being summed together. This step was intended to  
119 partially mimic the rectification and compression that is seen in the human auditory  
120 system [Plack et al., 2008]. The EEG and audio were subsequently downsampled to  
121 64 Hz and aligned in time using start-trigger events recorded with the EEG. EEG  
122 channels and audio data were Z-normalized to their mean and standard deviation  
123 in the training data.

## 124 **2.4 Cross-Validation Procedure**

125 The classifiers described in the following sections were trained and evaluated on  
126 data for each subject using a 10-fold nested cross-validation procedure. This ensures  
127 that the test data used to evaluate the classifiers is not used during any part of the  
128 training process (including hyperparameter tuning). The data were divided into 10  
129 folds and the outer cross-validation loop iterated over these folds. At each iteration,  
130 1 fold was held-out for testing, and the remaining 9 were used for training and  
131 hyperparameter tuning. Hyperparameters were tuned via an inner cross-validation  
132 loop: at each iteration of the inner loop, one fold was held out for validation and  
133 the remaining 8 were used for training. The objectives used for tuning the model  
134 hyperparameters are described with each model.

## 135 2.5 Stimulus-response models

136 Commonly-used stimulus-response models are shown in Figure 1. A forward stimulus-  
 137 response model predicts the EEG from the speech envelope, a backward model infers  
 138 the speech envelope from the EEG, and CCA maps both speech envelope and EEG  
 139 data into a common subspace. Here we consider only backward and CCA-based  
 140 models. The backward model, commonly used in decoding studies [Bialek et al.,  
 141 1991, Mesgarani et al., 2009, Mesgarani and Chang, 2012, Ding and Simon, 2012b,  
 142 Mirkovic et al., 2015, O’Sullivan et al., 2015, Van Eyndhoven et al., 2017, Wong  
 143 et al., 2018], serves as a baseline by which other models can be evaluated. The title  
 144 of the subsections describing each model (other than backward) or decoding scheme  
 145 contains a code in brackets, to make it easier to refer to various combinations of  
 146 these schemes.



**Figure 1:** Three main stimulus-response models. The forward model predicts the EEG from the speech envelope. The backward model infers (“reconstructs”) the speech envelope from the EEG. CCA projects both speech envelope and EEG data onto components in a common subspace. Correlation coefficients between predicted and actual EEG, inferred and actual stimulus, or canonical component (CC) pairs can be used as classification features.

### 147 2.5.1 Data format and notation

148 The audio stimulus envelope is represented as a matrix  $\mathbf{Y} = y_t$  of size  $T \times 1$  where  
 149  $T$  is the number of samples. The EEG signal is represented as a matrix  $\mathbf{X} = x_{t,n}$   
 150 of size  $T \times N$  where  $N$  is the number of channels. It may be useful to apply to  
 151 each channel a set of  $F$  time shifts, or process the each channel by a  $F$ -channel  
 152 filterbank. In that case  $\mathbf{X}$  designates the resulting matrix of size  $T \times FN$ .

## 153 2.5.2 Backward Model

154 Backward models have been used extensively for the AAD [Akram et al., 2016,  
155 Mirkovic et al., 2015, 2016, O’Sullivan et al., 2015, 2017] and match-vs-mismatch  
156 classification tasks [de Cheveigné et al., 2018, Di Liberto et al., In Review]. The  
157 backward model has been shown to result in better classification accuracy than the  
158 forward model for these tasks, as it permits a spatial filter to be applied to the EEG  
159 to take advantage of inter-channel covariance to filter out brain signals unrelated  
160 to the auditory cortical response [Wong et al., 2018]. Here, we extend this scheme  
161 to permit a *spatiotemporal* filter by augmenting the EEG data by applying a set of  
162 time lags. Time lagged data are concatenated along the channel dimension to form  
163 a matrix  $\mathbf{X}$  from which the audio envelope representation is inferred as  $\hat{\mathbf{Y}}$ :

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W} \quad (1)$$

164 The weights  $\mathbf{W}$  (spatiotemporal filter) are estimated from the training data us-  
165 ing ridge regression as in [Crosse et al., 2015, 2016, Holdgraf et al., 2017, O’Sullivan  
166 et al., 2017, Wong et al., 2018]:

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}, \quad (2)$$

167 where  $\lambda$  is the regularization parameter and  $\mathbf{I}$  is the identity matrix. The regular-  
168 ization parameter  $\lambda$  is optimized within the inner cross-validation loop to obtain the  
169 maximum correlation coefficient between the actual and predicted speech envelopes.  
170 An additional overall time shift parameter is also optimized within the inner loop.  
171 This time shift serves to absorb any latency mismatch due to filtering or cortical  
172 processing. The time-shift and  $\lambda$  parameters were optimized independently of each  
173 other, and in that order, for the purpose of saving time during model training.

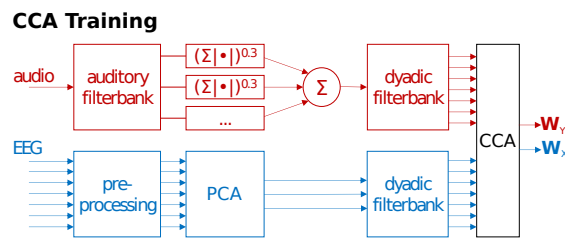
## 174 2.5.3 Canonical Correlation Analysis (C)

175 CCA finds linear transforms to apply to both audio and EEG to maximize mutual  
176 correlation. CCA has been shown to result in better classification accuracy than  
177 forward and backward models, as it allows spatiotemporal filters to be applied  
178 to both audio and EEG representations, stripping both of variance unrelated to  
179 the other [de Cheveigné et al., 2018]. CCA results in multiple pairs of canonical  
180 components (CCs), whereby the first has the largest correlation, and the second  
181 has the largest correlation that is orthogonal to the first, and so on. The audio  
182 and EEG CCs are computed as  $\mathbf{C}_Y = \mathbf{Y}\mathbf{W}_Y$  and  $\mathbf{C}_X = \mathbf{X}\mathbf{W}_X$ , respectively, where  
183  $\mathbf{Y}$  and  $\mathbf{X}$  are the audio and EEG data, and  $\mathbf{W}_Y$  and  $\mathbf{W}_X$  are the corresponding  
184 spatio-temporal CCA weights.

185 Time lags can be applied to the EEG (as previously described for the backward  
186 model) as well as the audio representation (as typically applied in forward models)

187 to allow the model to absorb convolutional mismatches between EEG and audio.  
 188 However, to capture long-range temporal structure would require many lags, leading  
 189 to computational issues and overfitting. For that reason, it is useful to replace the  
 190 time lags by a smaller number of filters [de Cheveigné et al., 2018]. Here we use a  
 191 set of  $F=9$  dyadic filterbank kernels that approximate a logarithmic filterbank. The  
 192 square-shaped left-aligned smoothing kernels have exponentially increasing lengths  
 193 from 1 to 32 samples. The resulting audio data matrix  $\mathbf{Y}$  has dimensions  $T \times$   
 194  $F$ , and the resulting EEG data  $\mathbf{X}$  has dimensions  $T \times NF$ , where the boxcar-  
 195 smoothing and EEG channel dimensions are combined into a single dimension.  
 196 Principal component analysis was applied to the filtered EEG data for whitening  
 197 and regularization. For regularization, principal components beyond a certain rank  
 198 were discarded before applying CCA. This is effectively a low rank approximation  
 199 (LRA) regularization scheme [Marconato et al., 2014]. The optimal number of EEG  
 200 principal components to keep was determined as the number that maximized the  
 201 cross-validated sum of correlation coefficients between CC pairs, over all pairs.

202 CCA was computed from the eigendecomposition of the covariance matrix  $\mathbf{R} =$   
 203  $([\mathbf{X}, \mathbf{Y}]^T [\mathbf{X}, \mathbf{Y}])$ , within the training dataset. The number of components,  $n_{cc}$  is  
 204 equal to the minimum size of the non-time dimension of  $\mathbf{X}$  or  $\mathbf{Y}$ . The CCA weights  
 205 for  $\mathbf{X}$ ,  $\mathbf{W}_X$ , are contained within the  $NF \times n_{cc}$  upper-left sub-matrix in  $\text{eig}(\mathbf{R})$ . Each  
 206 column of  $\mathbf{W}_X$  contains both channel and boxcar-smoothing dimensions, collapsed  
 207 into a single dimension. The CCA weights for  $\mathbf{Y}$ ,  $\mathbf{W}_Y$ , are contained within the  
 208  $F \times n_{cc}$  lower-left sub-matrix in  $\text{eig}(\mathbf{R})$ . An illustration of the CCA training inputs  
 209 and outputs is shown in Figure 2.



**Figure 2:** CCA training diagram. Preprocessed audio and EEG data are passed through a dyadic filterbank (see text). The CCA training algorithm then computes a set of weights  $\mathbf{W}_Y$  and  $\mathbf{W}_X$  that project the speech envelope and EEG data into a common subspace.

210 As for the backward model, an additional overall time shift parameter was in-  
 211 troduced to absorb any temporal mismatch between stimulus and response due to  
 212 filtering or cortical processing. This time-shift and the number of EEG principal  
 213 components retained (see above) were determined within the inner cross-validation  
 214 loop. They were determined independently and in that order to save computation.  
 215 Classification schemes that involve the CCA model are indicated with a code that



216 begins with the letter “C”.

## 217 2.6 Classification

218 The classification task is to decide, from a segment of EEG, which of two speech  
219 samples gave rise to it, the other being a sample from pseudorandom time window  
220 (“match vs mismatch” task). The features used for classification are, for the back-  
221 ward model, the correlation coefficient between the actual stimulus envelope and  
222 the estimate inferred from the EEG, and the correlation coefficient between the  
223 pseudorandom stimulus envelope and the estimate inferred from the EEG (bivari-  
224 ate feature). For the CCA model, the set of correlation coefficients between pairs  
225 of CCs is used (multivariate feature). The empirical joint distribution of features  
226 for matched and mismatched segments is estimated during the training phase of  
227 the classifier. For a new token containing an EEG segment paired with either the  
228 matching stimulus or a mismatching stimulus segment, the classifier identifies which  
229 of them corresponds to the match. Classification proceeds by situating the features  
230 relative to the empirical joint distribution for matched and mismatched pairs.

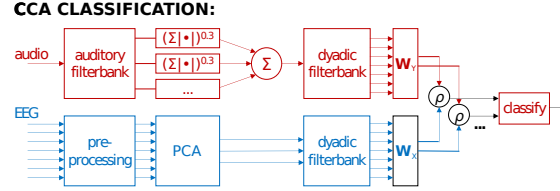
231 The classifier was trained anew on each iteration of the inner-cross-validation  
232 loop, using the model (backward or CCA) hyperparameters estimated on that it-  
233 eration. The optimal hyperparameters and classifier found over iterations of the  
234 inner loop were then applied to classify data within the left-out fold of the current  
235 iteration of the outer cross-validation loop. The average of classification scores over  
236 iterations of the outer loop are reported in the Results. To generate classification  
237 data samples, the position of the decoding segment was stepped by 1s throughout  
238 the evaluated data. The *decoding segment duration* was chosen among values 3, 5,  
239 7, 10 and 15s. These nominal durations include the length of the filtering kernels  
240 applied to the data (0.5s), as well as the optimal audio-EEG time-lag estimated in  
241 the hyperparameter estimation stage. Thus, they accurately reflect the duration of  
242 data used for each decision. The pseudorandom stimulus segment (foil) was drawn  
243 from a different fold from the actual speech sample. To allow reliable comparison  
244 between methods, the pseudorandom number generator was reinitialized with the  
245 same seed for the evaluation of each method.

246 For the backward model the classification feature was the correlation coefficient  
247 between the stimulus envelope and the envelope inferred from the EEG. To decode  
248 segment  $d$ , consisting of  $D$  time samples, the correlation coefficient between the  
249 predicted and actual speech envelope was computed as  $\rho_d = \frac{\hat{\mathbf{Y}}^T \mathbf{Y}}{\sqrt{\hat{\mathbf{Y}}^T \mathbf{Y} / D} \sqrt{\mathbf{Y}^T \mathbf{Y} / D}}$ .  
250 This feature was calculated for the stimulus segment within the test pair, and for a  
251 randomly chosen stimulus segment (foil). With this univariate feature, classification  
252 involves simply taking the larger correlation coefficient.

253 For the CCA model the classification feature was the set of correlation coef-



254 ficients between selected CC pairs (9 pairs in the implementation presented here,  
 255 since  $F = 9$ ), as illustrated in Figure 3. This feature was calculated for the stimulus  
 256 segment within the test pair, and for a randomly chosen stimulus segment (foil).  
 257 These two multivariate values were fed to a multivariate classifier. We consider lin-  
 258 ear discriminant analysis (next section) to obtain baseline classification rates, and  
 259 then proceed to neural network architectures.



**Figure 3:** CCA classification diagram. Preprocessed audio and EEG data are passed through a dyadic temporal filterbank, then projected via weights  $\mathbf{W}_Y$  and  $\mathbf{W}_X$  learned by CCA onto CCs. Correlation coefficients computed over a decoding segment duration between CC pairs are used as features for classifying whether one of two audio streams is the one that corresponds to the EEG data, or comes from a random segment of speech.

### 260 2.6.1 Linear Discriminant Analysis (L)

261 Denoting as  $x_i$  the multivariate correlation coefficient feature (for consistency with  
 262 standard expositions) and  $y_i$  the class label for token  $i$ , the predicted class is com-  
 263 puted as  $\hat{y}_i = \text{signum}(w \circ x_i)$ , where  $w$  is a weight vector and the  $y \in \{-1, +1\}$ .  
 264 LDA finds  $w$  such that the separation  $S$  between class distributions is maximized.  
 265  $S$  is defined as the ratio of the between class variance  $\sigma_b$  to the within class variance  
 266  $\sigma_w$ :

$$S = \frac{\sigma_b}{\sigma_w} = \frac{(w(\mu_{-1} - \mu_{+1}))^2}{w^T(\Sigma_{-1} + \Sigma_{+1})w}, \quad (3)$$

267 where  $\mu_{-1}$  and  $\mu_{+1}$  are the means of the two classes  $x_{i|y_i=-1}$  and  $x_{i|y_i=+1}$ , and  $\Sigma_{-1}$   
 268 and  $\Sigma_{+1}$  are their standard deviations.  $w$  can be found by solving the generalized  
 269 eigenvalue problem for the matrix  $S_w^{-1}S_b$ , where  $S_w$  is the within-class scatter matrix  
 270 and  $S_b$  is the between-class scatter matrix. Over all classes  $c$ , within-class scatter  
 271 matrix is given by  $S_w = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T$ . The between-class scatter  
 272 matrix is given by  $\sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T$ . The eigenvector corresponding to the largest  
 273 absolute eigenvalue is referred to as the first principal direction, or the weight vector  
 274  $w$ . The LDA classifier was trained on the correlation coefficients between the CCA-  
 275 transformed audio (actual and random) and the EEG. A classification scheme that  
 276 uses the LDA classifier is indicated by a code that ends in "L".

## 277 2.7 Improving Classification Rates

278 The methods described so far correspond to those used in a previous paper that  
279 compared forward, backward and CCA models associated with LDA [de Cheveigné  
280 et al., 2018]. In this section we introduce several schemes that go beyond those  
281 methods with the aim of improving classification rates. These are of two sorts:  
282 the first two schemes aim at obtaining better linear transform weight matrices than  
283 those produced by CCA, the last three schemes make use of neural net architectures  
284 to make better use of the features for classification.

### 285 2.7.1 D-Prime Maximization (D)

286 The cross-validation process described in Section 2.4 (inner loop) chooses hyperpa-  
287 rameters so as to obtain the highest possible *sum of correlation coefficients* between  
288 CC pairs. Large correlation coefficients scores on matched pairs (compared to mis-  
289 match) might be expected to lead to good discrimination, however discrimination  
290 also depends on *intra-class variance* of those coefficients [Wong et al., 2018]. This  
291 is captured by the d-prime sensitivity metric, calculated as the ratio between the  
292 inter-class means and intra-class variance. At each iteration of the inner cross-  
293 validation loop, a different set of CCA weights is computed for each regularization  
294 parameter sampled. By selecting those CCA weights that maximize the d-prime of  
295 output of a linear classifier applied to training data, classification error rates on the  
296 test set can potentially be reduced.

297 For each regularization parameter value sampled, the CCs computed on the val-  
298 idation data were divided into 2.5s segments. A classifier based on Kalman filtering  
299 was trained on the correlation coefficients between CC pairs for these segments.  
300 The derivation of this classifier permits a more analytic and stable evaluation of its  
301 d-prime score, although in practice it does not perform the match-vs-mismatch task  
302 as well as the LDA classifier. If we assume that the correlation coefficients between  
303 EEG and mismatching audio CCs have a mean of zero, and zero covariance with the  
304 correlation coefficients between EEG and matching audio CCs, the Kalman filter  
305 sensor matrix can be formulated as  $\mathbf{H} = \bar{\mathbf{Z}}_{match} - \bar{\mathbf{Z}}_{mismatch} = \bar{\mathbf{Z}}_{match}$  when the  
306 state  $y = 1$  and  $\mathbf{H} = \bar{\mathbf{Z}}_{mismatch} - \bar{\mathbf{Z}}_{match} = -\bar{\mathbf{Z}}_{match}$  when the state  $y = -1$ , where  
307  $\mathbf{Z}_{match} = \text{atanh}(\rho_{match})$  and  $\mathbf{Z}_{mismatch} = \text{atanh}(\rho_{mismatch})$ .  $\rho_{match}$  are the set of  
308 correlation coefficients between EEG and matching audio CCs. Similarly,  $\rho_{mismatch}$   
309 are the set of correlation coefficients between EEG and mismatching audio CCs.  
310 The hyperbolic-arctan-transform of is used to give  $\mathbf{Z}$  a Gaussian distribution. The  
311 Kalman gain can then be written as  $\mathbf{K} = \bar{\mathbf{Z}}^T [\bar{\mathbf{Z}}^T \bar{\mathbf{Z}} + \text{cov}(\mathbf{Z})]^{-1}$ , and the estimated  
312 states for each time sample in  $\mathbf{Z}$  is then  $\hat{y} = \tanh(\mathbf{Z} * \mathbf{K}^T)$ , given a previous neutral  
313 state of 0. The d-prime for the classifier output is thus expressed as  $d' = \frac{2\hat{y}}{\text{std}(\hat{y})}$ . For  
314 simplicity, the same data used to train the Kalman classifier was used to compute

315 the d-prime score.

316 Using an initial CCA regularization parameter value, an initial set of CCA  
317 weights was computed. The corresponding set of correlation coefficients between the  
318 resulting EEG and audio CCs, computed over 2.5s windows, was used to compute  
319 a Kalman classifier d-prime score. For each subsequent regularization parameter  
320 sampled, individual CCs were substituted into the previously established set and  
321 the d-prime score was recomputed. If an updated CC increased the d-prime score,  
322 the CCA weight corresponding to the updated CC was accepted as the new CCA  
323 weight. Abbreviated references to classification schemes implementing this method  
324 will include “*D*” in their name. For example, when CCA is applied using d-prime  
325 maximization and classification performed using LDA, this scheme will be denoted  
326 as “*CDL*”.

### 327 **2.7.2 Adaptive Beamforming (B)**

328 Given a training data set, CCA produces a set of spatiotemporal weight matrices  
329 that optimize correlation between CC pairs on the training data. The EEG weight  
330 matrix has two characteristics: (a) it preserves the useful brain activity that un-  
331 derlies the correlation and (b) it suppresses sources of noise that would otherwise  
332 degrade that correlation. When the trained solution is applied to new data, how-  
333 ever, the correlation structure of the noise may have changed so the solution is no  
334 longer optimal. The structure of the useful brain activity is less likely to change  
335 over time.

336 This situation can be addressed by applying a linearly constrained minimum  
337 variance (LCMV) beamformer. The LCMV beamformer, initially developed for  
338 antenna arrays, has proven useful to isolate localized neural activity by finding a  
339 weighted sum of EEG channels that project unit gain on a particular spatial lo-  
340 cation, while minimizing the contribution from all other locations. This type of  
341 beamforming is termed “adaptive” because the weights applied to the EEG chan-  
342 nels are adjusted to minimize the noise based on the covariance structure *of the data*  
343 *being analyzed*. The LCMV beamformer requires knowledge of the forward model  
344 of the desired source (source-to-sensor matrix). This is usually assumed to require  
345 computation from knowledge of the source position, together with a geometric de-  
346 scription of head tissues and tissue conductivity estimates, frequently taken from a  
347 structural MRI. However the formalism works just as well if the forward model is  
348 derived by other means. Here we derive it from the CCA solutions learned on the  
349 training set.

350 In this scenario, rather than corresponding to a specific spatial location, each  
351 “source” correspond to the forward model associated with a CC. Due to the orthog-  
352 onal nature of the CCA weights, the mapping from the CCs to the EEG sensors is  
353 computed from  $\mathbf{L} = [\text{eig}(\mathbf{R})^{-1}]^T$ , where  $\mathbf{R}$  is the covariance matrix used to compute

354 CCA from training data as described in Section 2.5.3. The first  $n_{cc}$  columns and  
355  $NF$  rows of  $\mathbf{L}$  correspond to the forward potentials of the  $n_{cc}$  CCs. We refer to this  
356 approach as “blind” in that it does not require knowledge of the actual geometry.

357 LCMV beamforming allows for the computation of weights that minimize noise  
358 within the EEG *test* data, and not just the training data. A forward model is  
359 derived from each of the CCs produced by applying CCA to the training data,  
360 based on which LCMV computes a beamforming weight vector that is used in lieu  
361 of the corresponding CC weight vector. In contrast to the CC weight vector that  
362 is fixed (after training) the beamforming weight vector is *adaptive*. This is useful  
363 in realtime applications where the nature of the noise is not always predictable,  
364 and also in batch processing of data with a complex non-uniform noise correlation  
365 structure.

366 In typical applications of LCMV to EEG data, such as neural source imaging,  
367 the forward potentials only contain a channel dimension, and sufficiently accurate  
368 forward potentials can be computed from a conductivity model of the head so that  
369 source localization can be performed. However, the CCA components yielded here  
370 contain both channel and boxcar-smoothing dimensions, combined into a single  
371 dimension. This larger dimensionality and the estimation of the source forward  
372 potentials from the data mean that these forward potentials are inexact. Errors in  
373 the forward potential can degrade beamformer performance, potentially resulting  
374 in the source of interest not being detected Dalal et al. [2014]. We use source  
375 suppression constraints to improve the solution, at the cost of reduced degrees  
376 of freedom for satisfying the beamforming objective of minimizing signal power.  
377 Given that each column in  $\mathbf{L}$  is uncorrelated with each other, and to each CC  
378 being measured, this relationship can be enforced in the beamformer solution by  
379 introducing them as source suppression constraints Dalal et al. [2006], Wong and  
380 Gordon [2009].

381 The typical LCMV beamformer constraints are 1) enforce unit gain on the EEG  
382 source corresponding to a given CCA component and 2) minimize signal power.  
383 These constraints yield the following beamformer equation [Van Veen et al., 1997]:

$$\mathbf{W}_{bf,X} = (\mathbf{L}_X^T \mathbf{R}_{test}^{-1} \mathbf{L}_X)^{-1} \mathbf{L}_X^T \mathbf{R}_{test}^{-1}, \quad (4)$$

384 where  $\mathbf{R}_{test}$  is the data covariance matrix computed in a similar way to  $\mathbf{R}$ , but  
385 over the validation or test fold.  $\mathbf{L}_X$  is the CCA forward potential column-vector,  
386 computed from the training data. Given that inaccuracies in the CCA forward  
387 potential estimate would result in reduced SNR and leakage from other sources, we  
388 add an additional constraint 3) enforce nulls on the EEG sources corresponding to  
389  $S$  uncorrelated sources, where  $S$  is optimized by cross-validation. This effectively  
390 minimizes the contribution of noise leakage into the beamformed signal. These  
391 uncorrelated source constraints are drawn from other columns in  $\mathbf{L}$ , which are or-

392 thogonal by definition. This third constraint is implemented by structuring  $\mathbf{L}_X$   
393 such that the first column is the forward potential corresponding to an individual  
394 CC being measured, and the remaining columns are the forward potentials of the  
395 sources to be suppressed. These columns are taken from the  $NF \times S$  upper-left  
396 sub-matrix in  $\mathbf{L}$ .

397 Given that a larger number of suppression constraints reduces the degrees of  
398 freedom available to the beamformer to suppress noise sources, the optimal number  
399 of suppression constraints  $S$  needs to be determined. This is done via the 9-fold  
400 inner cross-validation described in Section 2.4.  $S$  was determined separately for  
401 each CC. Thus, the beamforming implementation with CCA effectively involves  
402 tuning three types of regularization parameters to maximize the cross-validated  
403 sum of correlation coefficients across CC pairs: the number of lags, the number of  
404 principal components kept when whitening EEG, and the number of suppression  
405 constraints per CC. The number of lags is determined first, independent of the  
406 others. The number of principal components to keep and the number of suppression  
407 constraints are then determined via a grid search. Note that here as with the default  
408 CCA implementation, the same number of principal components is kept for all CCs.  
409 Classification schemes implementing this method will include “ $B$ ” in their name.

410 Beamforming and d-prime maximization can be combined. With d-prime maxi-  
411 mization and no beamforming, while adjusting the number of principal components  
412 kept during EEG whitening as a regularization parameter, individual CCA weights  
413 that maximized the validation classifier d-prime were kept. When combined with  
414 beamforming, rather than keeping the individual CCA weights, the individual CCA  
415 forward potentials and associated source suppression constraints are kept instead.

### 416 2.7.3 Multilayer Perceptron (M)

417 The LDA classifier uses only the principal direction in multivariate space to sep-  
418 arate the two classes. Other directions, possibly also informative for class sep-  
419 aration, are ignored. A multilayer perceptron (MLP) neural network can find a  
420 nonlinear decision function that may be better as it combines information from  
421 multiple decision planes. We implemented a multilayer perceptron (MLP) neu-  
422 ral network with hyperbolic tangent activation functions, feeding into a two-unit  
423 softmax classification layer. An MLP layer performs a nonlinear operation on the  
424 inputs  $y_i = \tanh(\mathbf{W}x_i + b)$ , where  $\mathbf{W}$  is the weight matrix and  $b$  is the bias vec-  
425 tor. Multiple MLP layers can be stacked so that subsequent layers take the output  
426 from the previous layer as input. A softmax layer takes the output from the last  
427 MLP layer as its input and computes an output such that each value  $c$  in  $y_i$ , cor-  
428 responding to each class, is normalized according to  $y_{i,c} = \frac{e^{x_i^T \mathbf{w}_c}}{\sum_j^C e^{x_i^T \mathbf{w}_j}}$ . The largest  
429 of the values  $c$  in  $y_i$  corresponds to the predicted class. The network was trained  
430 using a categorical cross-entropy cost function. Training was performed using mini-

431 batches, and *rmsprop* as the gradient descent method [Hinton et al., 2012]. Early  
432 stopping was used to terminate training when the validation cost function no longer  
433 improved. Different numbers of MLP layers and units per layer were experimented  
434 with. Abbreviated references to the CCA classification schemes using an MLP will  
435 end in “*M*”.

#### 436 2.7.4 Simple Recurrent Layer (S)

437 Up to this point, single correlation coefficients have been computed over the entire  
438 segment of data used for classification. A correlation coefficient is the dot product  
439 between two normalized CC time series in which all time points are weighted equally.  
440 However, if information could be obtained as to which time points are more reliable,  
441 it would be more appropriate to apply a non-uniform weight to modulate the amount  
442 each time point contributes to the final classification. We divided the correlation  
443 coefficient computation over the entire decoding segment into non-overlapping sub-  
444 intervals of one second duration. The number of sub-intervals was thus equal to  
445 the decoding segment duration in seconds. Based on the correlation coefficient  
446 equation, the *sub-interval correlation coefficient* computed over a sub-interval  $s$  is  
447 computed as:

$$\rho_s = \text{diag} \left[ \left( \frac{\mathbf{C}_{X,s}}{\sqrt{\sum_t \mathbf{C}_X^2/D}} \right)^T \left( \frac{\mathbf{C}_{Y,s}}{\sqrt{\sum_t \mathbf{C}_Y^2/D}} \right) \right], \quad (5)$$

448 where for a total of  $S$  sub-intervals,  $\mathbf{C}_{X,s} \equiv \mathbf{C}_X(t \in (s, s + 1]D/S)$ , and similarly  
449  $\mathbf{C}_{Y,s} \equiv \mathbf{C}_Y(t \in (s, s + 1]D/S)$ . This denominators of this equation are computed  
450 over the entire decoding segment duration, whereas the numerators are computed  
451 only over the sub-interval. Since the denominator can be seen as a normalization  
452 factor, computing  $\rho_s$  in this way stabilizes the normalization factor over the entire  
453 interval.

454 To determine the weighting for each sub-interval, we chose to employ a simple  
455 recurrent network (SRN) layer which takes only the sub-interval correlation coeffi-  
456 cients as inputs. An SRN takes a set of input vectors over  $S$ -time steps,  $x_s$ . For each  
457 time step, it computes a new state  $h_s$  based on its previous state  $h_{s-1}$  and input  
458 vector  $x_s$  according to  $h_s = \tanh(\mathbf{W}_h x_s + \mathbf{U}_h h_{s-1} + b_h)$ . The output of the last SRN  
459 timestep was passed to a 2-layer MLP, consisting of 3 units each, terminating in a  
460 softmax output layer for classification. Training was performed using minibatches,  
461 and *rmsprop* as the gradient descent method [Hinton et al., 2012]. Abbreviated  
462 references to the CCA classification scheme using an SRN will end with “*S*”.



### 463 2.7.5 Gated Recurrent Unit (G)

464 An SRN lacks the ability to store information over long durations due to the vanish-  
465 ing gradient problem: the SRN time-steps are unfolded into a multi-layer network  
466 for training, and with the use of sigmoid-like activation functions, the backpropa-  
467 gated error diminishes across layers, preventing the SRN from learning long-term  
468 relationships [Pascanu et al., 2013]. In contrast, a gated recurrent unit (GRU) al-  
469 lows the error to be preserved through time and layers. A GRU updates its internal  
470 state  $h_s$  based on two gating functions: the update gate  $z_s$  and the reset gate  $r_t$   
471 [Cho et al., 2014]. The update gate determines how much of the current state  
472  $h_s$  at timestep  $s$  incorporates the previous state  $h_{s-1}$  versus a candidate state  $\tilde{h}_s$   
473 computed from  $x_s$  plus some leakage from  $h_{s-1}$ .

$$h_s = (1 - z_s) \circ h_{s-1} + z_s \circ \tilde{h}_s \quad (6)$$

474 The update gate  $z_t$  is computed as a sigmoid function of the weighted GRU  
475 input  $x_t$  and the previous state  $h_{t-1}$ :

$$z_s = \sigma(\mathbf{W}_z x_s + \mathbf{U}_z h_{s-1} + b_z), \quad (7)$$

476 where  $\mathbf{W}_z$  and  $\mathbf{U}_z$  are weights and  $b_z$  is a bias vector. The candidate state  $\tilde{h}_s$  is  
477 computed as a hyperbolic tangent function of the weighted GRU input  $x_s$  and the  
478 weighed previous state  $h_{s-1}$ :

$$\tilde{h}_s = \tanh(\mathbf{W}_h x_s + r_s \circ \mathbf{U}_h h_{s-1} + b_h), \quad (8)$$

479 where  $\mathbf{W}_h$  and  $\mathbf{U}_h$  are weights and  $b_h$  is a bias vector. The reset gate,  $r_s$  determines  
480 how much leakage from the previous state is incorporated into  $h_s$ . Similar to the  
481 update gate, is computed as a function of weighted GRU input and the previous  
482 state.

$$r_s = \sigma(\mathbf{W}_r x_s + \mathbf{U}_r h_{s-1} + b_r), \quad (9)$$

483 where  $\mathbf{W}_r$  and  $\mathbf{U}_r$  are weights and  $b_r$  is a bias vector.

484 The GRU layer consisted of 8 units. The output of the last GRU timestep  
485 was passed to a 2-layer MLP, consisting of 3 units each, terminating in a softmax  
486 output layer for classification. Abbreviated references to the CCA classification  
487 scheme using an GRU will end with “G”.

## 488 2.8 Classifier Performance Evaluation

489 We used two metrics to quantify performance: classification error rate and infor-  
490 mation transfer rate (ITR). The ITR is the number of correct decisions that can be  
491 made by the classifier per unit time. Because increased decoding segment lengths

492 result in a reduction in the number of decisions that can be made per unit time,  
493 this measure allows for the comparison of results across different decoding seg-  
494 ment lengths. The ITR measure that was used was the Wolpaw ITR [Wolpaw and  
495 Ramoser, 1998] and is calculated by:

$$ITR_W = V \left[ \log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1} \right], \quad (10)$$

496 where  $V$  is the speed in trials per minute,  $N$  is the number of classes, and  $P$  is the  
497 classifier accuracy (1 minus the error rate). Both metrics were averaged across all  
498 test folds for each subject.

## 499 **2.9 Implementation**

500 Data preprocessing and CCA analyses were performed using the COCOHA Matlab  
501 Toolbox [Wong et al., 2018]. The scikit-learn implementation of LDA was used,  
502 and the neural networks were implemented in Keras [Chollet et al., 2015].

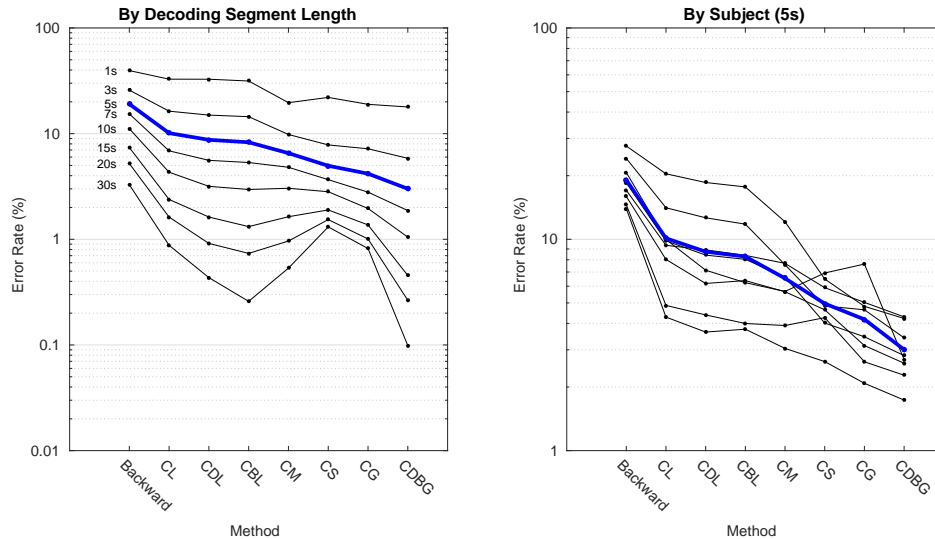
## 503 **3 Results**

504 A summary of the classification performance of all methods is shown in Figure 4.  
505 Performance is quantified here as percent *error rate* rather than percent correct  
506 rate as is common: lower is better. The left panel shows the average error rate over  
507 subjects for a range of decoding segment lengths, and the right panel shows the  
508 error rate at a 5s decoding segment length for each subject. Moving left to right,  
509 a clear improvement can be seen as new methods are introduced and combined.  
510 Taking the backward model as a baseline, the best combination reduces the error  
511 rate by from 18.9% to 3.0% (i.e. by a factor of 6.3). The contribution of each step  
512 is detailed in the following. For paired t-test analyses of error rate data, a logit  
513 transform is applied to the error rates [Warton and Hui, 2011].

### 514 **3.1 CCA vs backward model**

515 CCA+LDA ( $CL$ ) provides a clear improvement over the backward model, as we  
516 found previously [de Cheveigné et al., 2018]. At a decoding segment length of 5s  
517 the error rate decreased by 9.0 percentage points (paired samples t-test,  $T_{79} = 21.9$ ,  
518  $p = 2.7 \times 10^{-35}$ ), that is by a factor of 1.89, on average over subjects. The difference  
519 is of same sign for all subjects, and all durations. It is instructive to see how this  
520 improvement relates to the original error rate.

521 Figure 5 left shows a scatterplot of error rates for the CCA+LDA scheme vs  
522 the backward model. Each dot represents the error rate for one test fold and one  
523 subject. The axes are scaled by a logit transform to account for the saturation effect



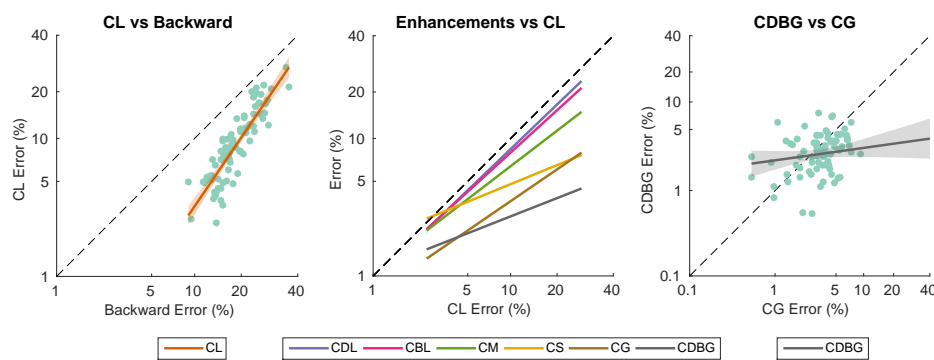
**Figure 4:** Classification error rate for different classification schemes. Chance is 50%. The left panel shows error rates with different decoding segment lengths, averaged over all subjects. The right panel shows error rates for each subject using a 5s decoding segment length. The average of all subjects is indicated by the blue line, which also corresponds to the blue line in the left panel.  $C = \text{CCA}$ ,  $D = \text{d-prime maximization}$ ,  $B = \text{beamforming}$ ,  $L = \text{LDA}$ ,  $M = \text{MLP}$ ,  $S = \text{SRN}$ ,  $G = \text{GRU}$ .

524 as the error rate decreases to 0 [Warton and Hui, 2011]. This transform produces a  
 525 normal distribution and equivariance in regression residuals, which are underlying  
 526 assumptions of linear regression model statistics. On these axes the data follow  
 527 a linear trend with slope  $m = 1.47$  greater than one ( $CI_{.95} = [1.29, 1.66]$ ). This  
 528 indicates that the benefit was greater for classification folds that already had a low  
 529 error rate, after accounting for the effects of saturation.

530 We now use the CCA+LDA model ( $CL$ ) as a baseline to evaluate schemes that  
 531 further improve performance. We report the effect of scheme is shown in isolation  
 532 (relative to  $CL$ ) as well as their best-performing combination ( $CDBG$ ). We also  
 533 analyze improvement as a function of the baseline error rate, as summarized in  
 534 Figure 5 (center).

### 535 3.2 D-Prime Maximization ( $D$ )

536 D-prime maximization (see Methods) yielded a 1.4 percentage point (a factor of  
 537 1.16) classification error decrease (paired samples t-test,  $T_{79} = 5.9$ ,  $p = 6.6 \times 10^{-8}$ ).  
 538 The purple line in Figure 5 (center) represents a linear fit of the scatter plot error  
 539 rates of  $CDL$  vs  $CL$ . The slope  $m = 0.96$  was not significantly different from 1. In  
 540 other words, maximizing the d-prime scores equally reduces the classification error  
 541 of all folds regardless of the original CCA+LDA classifier ( $CL$ ) error.



**Figure 5:** Error rate relationships between classification schemes, with lines of best fit. The graphs are plotted on logit axes in order to compensate for effects of saturation as error rates approach 0 or 100% [Warton and Hui, 2011]. The left panel shows the error rate of CCA+LDA ( $CL$ ) versus that of the backward model. The center panel shows linear fits to scatter plots of error rates of several enhanced schemes relative to CCA+LDA ( $CL$ ). The right panel shows a scatterplot of the error rate for the best combination ( $CDBG$ ) relative to the second best ( $CG$ ). Dots in left and right panels represent classification errors for test folds, shown for all subjects. Translucent bands around the lines in the left and right panels indicate the 95% confidence intervals.  $C$  = CCA,  $D$  = d-prime maximization,  $B$  = beamforming,  $M$  = MLP,  $S$  = SRN,  $G$  = GRU.

### 542 3.3 Beamforming ( $B$ )

543 Beamforming (see Methods) yielded a 1.8 percentage point (a factor of 1.22) clas-  
 544 sification error decrease (paired samples t-test,  $T_{79} = 7.1$ ,  $p = 4.8 \times 10^{-10}$ ). The  
 545 red line in Figure 5 (center) represents a linear fit to the scatterplot of error rates  
 546 of ( $CBL$ ) versus  $CL$ . The slope  $m = 0.91$  was not significantly different from 1.

### 547 3.4 Multilayer Perceptron ( $M$ )

548 A four-layer MLP network, with 8 units in the first layer and 3 units second and third  
 549 layers, followed by a 2-unit softmax classification layer, achieved an 3.6 percentage  
 550 point (a factor of 1.55) classification error decrease over the original CCA+LDA  
 551 classifier ( $CL$ ) (paired samples t-test,  $T_{79} = 12.3$ ,  $p = 5.2 \times 10^{-20}$ ). The green line  
 552 in 5 (center) represents a linear fit of the scatterplot of error rates of  $CM$  vs  $CL$ .  
 553 The slope of  $m = 0.77$  was significantly less than 1 ( $CI_{.95} = [0.66, 0.88]$ ), indicating  
 554 that the error rate decreased more for folds that had larger error rates. Increasing  
 555 the number of layers or units per layer did not significantly impact the classification  
 556 performance.

### 557 **3.5 Simple Recurrent Network (*S*)**

558 Replacing the first MLP layer with an 8-unit simple recurrent network (SRN)  
559 achieved a 5.2 percentage point (a factor of 2.04) classification error decrease (paired  
560 samples t-test,  $T_{79} = 11.9$ ,  $p = 2.4 \times 10^{-19}$ ). The yellow line in Figure 5 (center)  
561 represents a linear fit of the scatterplot of error rates of *CS* vs textitCL. The slope  
562  $m = 0.41$  was significantly less than 1 ( $CI_{.95} = [0.26, 0.56]$ ), indicating that the  
563 error decreased more for folds that had larger error rates.

564 It is worth noting that MLP and SRN classifiers perform less well at longer  
565 durations (Figure 4, left), and at 20 and 30s the SRN classifier (*CS*) yields greater  
566 error rates the original *CL* scheme. This is possibly a result of the vanishing gradient  
567 problem which prevents the SRN from learning long-term relationships, and thereby  
568 impedes performance when the recurrent classifier must make a prediction after  
569 processing a larger number of sub-intervals.

### 570 **3.6 Gated Recurrent Unit (*G*)**

571 Replacing the SRN layer by an 8 unit GRU layer yielded a 5.9 percentage point  
572 (a factor of 2.42) classification error decrease (paired samples t-test,  $T_{79} = 15.8$ ,  
573  $p = 4.3 \times 10^{-26}$ ). The brown line in Fig. 5 (center) represents a linear fit of the  
574 scatterplot of error rates of *CG* vs (textitCL. The slope  $m = 0.68$  was significantly  
575 less than 1 ( $CI_{.95} = [0.47, 0.89]$ ). Again this indicates that the error decreased more  
576 for folds that had larger error rates, although folds with small error rates also seem  
577 to benefit (Figure 5 center).

578 The classifier with a GRU layer (*CG*) performed better than a classifier with a  
579 SRN layer (*CG*, (paired samples t-test,  $T_{79} = 4.4$ ,  $p = 3.8 \times 10^{-5}$ ). To determine  
580 whether this could be due to the larger number of parameters used in the GRU  
581 network (693 parameters, including the MLP portion), we implemented also a clas-  
582 sifier with an SRN layer with 17 units (684 parameters). The classification error for  
583 the larger SRN was significantly larger than that obtained by the 8-unit SRN by  
584 1.3 percentage points (paired samples t-test,  $T_{79} = 7.2$ ,  $p = 3.1 \times 10^{-10}$ ) suggesting  
585 overfitting. The advantage of the GRU is thus unlikely to be related to its larger  
586 number of parameters.

### 587 **3.7 Combined Methods (*CDBG*)**

588 The GRU (*CG*) yielded the largest decrease in error rate over the basic CCA+LRA  
589 implementation for durations up to 10s (Figure 4 left). However, combining it  
590 with several of those schemes yielded a yet greater improvement (*CDBG*). Adding  
591 d-prime maximization and beamforming reduced the error rate by 1.2 percentage  
592 points (paired samples t-test,  $T_{79} = 2.49$ ,  $p = 0.015$ ), that is a factor of 1.39.

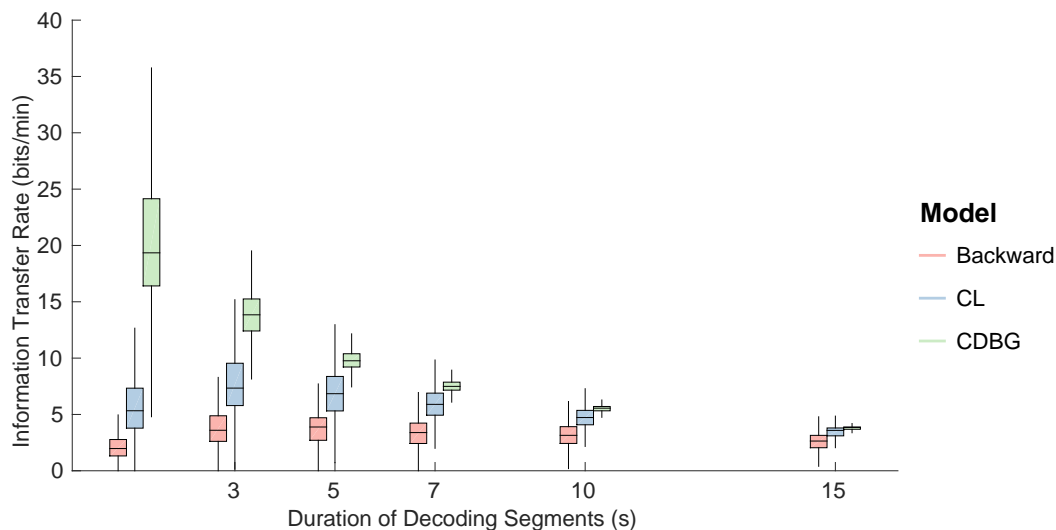
593 Interestingly, this benefit extended also to long durations (Figure 4 left), attaining  
594 an error rate of 0.1% for 30s duration segments (compared to 3% for the backward  
595 model).

596 Figure 5 (right) shows a scatterplot of error rates for *CDBG* relative to *CG*. The  
597 slope  $m = 0.14$  is significantly smaller than 1 ( $CI_{.95} = [-0.04, 0.34]$ ), indicating that  
598 the improvement is greatest for folds/subjects for which error rates were relatively  
599 high.

### 600 3.8 Information Transfer Rate (ITR)

601 From Figure 4 (left) it is obvious that there is a tradeoff between error rate and  
602 segment duration, shorter segments yielding greater error rates. An alternative  
603 metric of performance is ITR (roughly, the number of decisions that can be made  
604 per unit time, see Methods). Such a metric is relevant for BCI applications that  
605 require decisions to be both accurate and timely.

606 Figure 6 plots values of the ITR for the backward model (red), *CCA+LDA*  
607 (*CL*, blue), and *CCA* with d-prime maximization, beamforming, and GRU neural  
608 network improvements (*CDBG*, green). As expected from the error rate metric, the  
609 more sophisticated schemes yield higher ITR rates. The maximum ITR is reached  
610 at 5s for the backward model, 3s for *CL* and 1s for *CDBG*).



**Figure 6:** Information transfer rate comparison over different decoding segment durations for the backward model, the baseline *CCA* implementation using an LRA classifier (*CL*), and the *CCA* implementation with all enhancements: d-prime maximization, beamforming and a GRU classifier (*CDBG*).



## 611 4 Discussion

612 In previous work we found that a CCA-based model yielded more accurate clas-  
613 sification than standard backward or forward models [de Cheveigné et al., 2018],  
614 presumably thanks to the ability of CCA to factor out irrelevant information from  
615 both audio and EEG, and to provide multiple components to support multivariate  
616 classification. In the present paper, we observed that the benefit over the backward  
617 model (in relative terms) was smaller for folds where the backward model gave  
618 larger classification errors (Figure 5 left), suggesting that performance might be  
619 limited by poor EEG data quality on those folds. Thus, we focused on improving  
620 the CCA classification framework to be more robust to noise (defined as any feature  
621 of the data that increases classification error). This encompasses EEG artifacts, but  
622 it might also include points in the audio that do not yield reliable EEG response,  
623 such as silences. The solutions explored were methods to improve estimation of the  
624 CCA weights, and to allow a classifier to utilize temporal information.

### 625 4.1 Improving CCA Weights

626 When applied to CCA+LDA, both d-prime maximization and beamforming reduced  
627 classification errors equally across classification folds, regardless of the original error.  
628 Maximization of component d-prime yielded EEG spatial filter weights that were  
629 superior to those provided by CCA. This operation was performed individually for  
630 each CC. An alternative approach, not explored in the present study, could be to  
631 maximize the d-prime output or the loss function of a classifier via tuning of all  
632 components in combination.

633 Beamforming is another approach to improve spatial filter weights. It requires  
634 knowledge of the forward potentials of sources to preserve. Typically this knowledge  
635 is computed from anatomical data and models of head tissue conductivity, but  
636 here we use forward potentials associated with optimal components computed from  
637 CCA. Beamforming adaptively suppresses activity other than that associated with  
638 the forward potentials, effectively addressing the time-varying structure of the noise.  
639 We did not make full use of this flexibility in our simulations: beamforming was  
640 applied on the basis of the covariance matrix calculated over the full length of the  
641 cross-validation fold, which is roughly 9 minutes. An alternative, not explored in the  
642 present study, is to recalculate the beamformer solution based on shorter intervals.  
643 There is, however, a limit to which the time window can be shortened as sufficient  
644 data is needed to accurately estimate the covariance matrix  $\mathbf{R}$ .

## 645 **4.2 Improving the Classifier**

646 A multilayer perceptron (MLP) network reduced classification errors slightly com-  
647 pared to an LDA classifier, suggesting that there is some advantage that can be  
648 gained from a nonlinear decision function. However, the recurrent neural networks  
649 (SRN and GRU) showed the largest reduction in classification error over an LDA  
650 classifier. The recurrent networks yielded the greatest benefit for folds with higher  
651 CCA+LDA classification errors, suggesting that they can tackle noise features for  
652 which the other classifiers fail. The recurrent layers are likely able to handle shorter-  
653 term variations in the noise, compared to d-prime maximization or beamforming,  
654 that are calculated over the entire cross-validation/test dataset. The time-scale of  
655 variations in the noise that can be handled by the SRN or GRU are related to the  
656 length of the sub-intervals used to compute the correlation coefficients fed to these  
657 neural network layers. While the GRU provided the largest reduction in classifi-  
658 cation error over CCA+LDA, combining it with component d-prime maximization  
659 and beamforming provided a significant additional reduction.

## 660 **4.3 Relation between same-different and AAD tasks**

661 The results reported in this paper were obtained for a match-vs-mismatch classifi-  
662 cation task, that allowed us to focus on the quality of the stimulus-response model.  
663 We preferred this task to the more complex AAD task, as it is not vulnerable to  
664 mislabeling of the database. In the AAD task an “error” might be the result of  
665 attention drift, making it hard to explore the performance in the region of low error  
666 rates (of use for applications). Cortical responses to concurrent speakers have been  
667 shown to have slightly different dynamics than those to a single speaker. [Ding  
668 and Simon, 2012b] found that the attended talker shows a stronger representation  
669 than the unattended talker at longer latencies ( $\approx 200\text{ms}$ ), whereas both attended  
670 and unattended talkers are equally represented at shorter latencies ( $\approx 80\text{ms}$ ). We  
671 expect our methods to be effective also in the AAD task, but it would be useful to  
672 confirm this in future studies.

673 Extrapolating from our results, and considering the many paths that remain to  
674 be explored, we believe that further improvements may be possible.

## 675 **4.4 Summary**

676 Previous studies showed that the relation between stimulus and brain response can  
677 be captured by a linear model fit using system identification techniques, extending  
678 classic ERP studies to allow continuous stimuli such as speech [Lalor et al., 2006,  
679 2009, Lalor and Foxe, 2010, Power et al., 2012]. Such a linear model can be used  
680 by a classifier in a BCI application, for example to decide whether a listener is at-

681 tending to one or the other of two concurrent voices (AAD), but poor classification  
682 reliability and the amount of data required by each decision limit the practical use  
683 of such a scheme [O’Sullivan et al., 2017, Zink et al., 2017, Wong et al., 2018].  
684 In previous work we showed that the stimulus-response model can be significantly  
685 improved using CCA [de Cheveigné et al., 2018], and here we showed that classifica-  
686 tion performance can be further enhanced by improving the quality of EEG linear  
687 filters over CCA, or improving the classifier over LDA. Overall, the error rate was  
688 divided by 6 over the standard backward model, for a 5s segment of data. This  
689 brings us closer to the goal of reliable “cognitive control” of a device based on brain  
690 responses.

## 691 5 Acknowledgements

692 This work was supported by the EU H2020-ICT grant 644732 (COCOHA), and  
693 grants ANR-10-LABX-0087 IEC and ANR-10-IDEX-0001-02 PSL. It draws on work  
694 performed at the 2016 Telluride Neuromorphic Engineering workshop.

## 695 References

- 696 S. Akram, A. Presacco, J.Z. Simon, S.A. Shamma, and B. Babadi. Robust decoding  
697 of selective auditory attention from meg in a competing-speaker environment via  
698 state-space modeling. *Neuroimage*, 124(Pt A):906–917, 2016.
- 699 W. Bialek, F. Rieke, R.R. de Ruyter Van Steveninck, and D. Warland. Reading a  
700 neural code. *Science*, 252(5014):1854–1857, 1991.
- 701 M.P. Broderick, A.J. Anderson, G.M. Di Liberto, M.J. Crosse, and E.C. Lalor.  
702 Electrophysiological correlates of semantic dissimilarity reflect the comprehension  
703 of natural, narrative speech. *Curr. Biol.*, 28(5):803–809.e3, 2018a.
- 704 M.P. Broderick, A.J. Anderson, G.M. Di Liberto, M.J. Crosse, and E.C.  
705 Lalor. Data from: Electrophysiological correlates of semantic dissimilar-  
706 ity reflect the comprehension of natural, narrative speech, 2018b. URL  
707 <https://doi.org/10.5061/dryad.070jc>.
- 708 K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk,  
709 and Y. Bengio. Learning phrase representations using RNN encoder-decoder for  
710 statistical machine translation. *arXiv*, 2014.
- 711 François Chollet et al. Keras. <https://keras.io>, 2015.

- 712 M.J. Crosse, J.S. Butler, and E.C. Lalor. Congruent visual speech enhances cortical  
713 entrainment to continuous auditory speech in noise-free conditions. *J. Neurosci.*,  
714 35(42):14195–14204, 2015.
- 715 M.J. Crosse, G.M. Di Liberto, A. Bednar, and E.C. Lalor. The multivariate tempo-  
716 ral response function (mTRF) toolbox: a MATLAB toolbox for relating neural  
717 signals to continuous stimuli. *Front. Hum. Neurosci.*, 10:604, 2016.
- 718 S.S. Dalal, K. Sekihara, and Nagarajan S.S. Modified beamformers for coherent  
719 source region suppression. *IEEE Trans. Biomed. Eng.*, 53(7):1357–63, 2006.
- 720 S.S. Dalal, S. Rampp, F. Willomitzer, and S. Ettl. Consequences of eeg electrode  
721 position error on ultimate beamformer source reconstruction performance. *Front.*  
722 *Neurosci.*, 8(42), 2014. doi: 10.3389/fnins.2014.00042.
- 723 A. de Cheveigné and L. Parra. Joint decorrelation: a versatile tool for multichannel  
724 data analysis. *Neuroimage*, 98:487–505, 2014.
- 725 A. de Cheveigné, D.D.E. Wong, G.M. Di Liberto, J. Hjortkjær, M. Slaney, and  
726 E. Lalor. Decoding the auditory brain with canonical component analysis. *Neu-*  
727 *roimage*, 172:206–216, 2018.
- 728 G.M. Di Liberto, D.D.E. Wong, G.A. Melnik, and de Cheveigné. Cortical responses  
729 to natural speech reflect probabilistic phonotactics. *J. Neurosci.*, In Review.
- 730 N. Ding and J.Z. Simon. Neural coding of continuous speech in auditory cortex  
731 during monaural and dichotic listening. *J. Neurophysiol.*, 107(1):78–89, 2012a.  
732 doi: 10.1152/jn.00297.2011.
- 733 N. Ding and J.Z. Simon. Emergence of neural encoding of auditory objects while  
734 listening to competing speakers. *Proc. Natl. Acad. Sci. U. S. A.*, 109(29):11854–  
735 11859, 2012b. doi: 10.1073/pnas.1205381109.
- 736 N. Ding and J.Z. Simon. Adaptive temporal encoding leads to a background-  
737 insensitive cortical representation of speech. *J. Neurosci.*, 33(13):5728–5735, 2013.
- 738 N. Ding and J.Z. Simon. Cortical entrainment to continuous speech: functional  
739 roles and interpretations. *Front. Hum. Neurosci.*, 8, 2014.
- 740 J.P. Dmochowski, J.J. Ki, P. DeGuzman, P. Sajda, and L.C. Parra. Ex-  
741 tracting multidimensional stimulus-response correlations using hybrid encoding-  
742 decoding of neural activity. *Neuroimage*, 180(PtA):134–146, 2017. doi:  
743 10.1016/j.neuroimage.2017.05.037.
- 744 S.A. Hillyard, R. F. Hink, V. L. Schwent, and T. W. Picton. Electrical signs of  
745 selective attention in the human brain. *Science*, 182(108), 1973.

- 746 G. Hinton, N. Srivastava, and K. Swersky. Lecture notes in neural networks for  
747 machine learning, 2012.
- 748 C.R. Holdgraf, J.W. Rieger, C. Micheli, S. Martin, R.T. Knight, and F.E. Theunis-  
749 sen. Encoding and decoding models in cognitive electrophysiology. *Front. Sys.*  
750 *Neurosci.*, 11:61, 2017.
- 751 H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.
- 752 E. C. Lalor, A. J. Power, R. B. Reilly, and J. J. Foxe. Resolving precise tem-  
753 poral processing properties of the auditory system using continuous stimuli. *J.*  
754 *Neurophysiol.*, 102(1):349–59, 2009.
- 755 E.C. Lalor and J.J. Foxe. Neural responses to uninterrupted natural speech can  
756 be extracted with precise temporal resolution. *Eur. J. Neurosci.*, 31(1):189–193,  
757 2010. doi: 10.1111/j.1460-9568.2009.07055.x.
- 758 E.C. Lalor, B.A. Pearlmutter, R.B. Reilly, G. McDarby, and J.J. Foxe. The VESPA:  
759 a method for the rapid estimation of a visual evoked potential. *Neuroimage*, 32  
760 (4):1549–1561, 2006.
- 761 A. Marconato, L. Ljung, Y. Rolain, and J. Schoukens. Linking regularization and  
762 low-rank approximation for impulse response modeling. *IFAC Proc. Vol.*, 47(3):  
763 4999–5004, 2014.
- 764 N. Mesgarani and E.F. Chang. Selective cortical representation of attended speaker  
765 in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012.
- 766 N. Mesgarani, S.V. David, J.B. Fritz, and S.A. Shamma. Influence of context and  
767 behavior on stimulus reconstruction from neural activity in primary auditory  
768 cortex. *J. Neurophysiol.*, 102(6):3329–3339, 2009.
- 769 B. Mirkovic, S. Debener, M. Jaeger, and M. De Vos. Decoding the attended speech  
770 stream with multi-channel EEG: implications for online, daily-life applications.  
771 *J. Neural Eng.*, 12(4):046007, 2015. doi: 10.1088/1741-2560/12/4/046007.
- 772 B. Mirkovic, M.G. Bleichner, M. De Vos, and S. Debener. Target speaker detec-  
773 tion with concealed EEG around the ear. *Front. Neurosci.*, 10:349, 2016. doi:  
774 10.3389/fnins.2016.00349.
- 775 J.A. O’Sullivan, A.J. Power, N. Mesgarani, S. Rajaram, J.J. Foxe, B.G. Shinn-  
776 Cunningham, M. Slaney, S.A. Shamma, and E.C. Lalor. Attentional selection  
777 in a cocktail party environment can be decoded from single-trial EEG. *Cereb.*  
778 *Cortex*, 25(7):1697–1706, 2015. doi: 10.1093/cercor/bht355.

- 779 J.A. O’Sullivan, Z. Chen, J. Herrero, G.M. McKhann, S.A. Sheth, A.D. Mehta,  
780 and N. Mesgarani. Neural decoding of attentional selection in multi-speaker  
781 environments without access to clean sources. *J. Neural Eng.*, 14(5):056001,  
782 2017.
- 783 R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent  
784 neural networks. *Proc. Mach. Learn Res.*, 28(3):1310–1318, 2013.
- 785 R.D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice. An efficient auditory  
786 filterbank based on the gammatone function. In *Meeting of the IOC Speech Group  
787 on Auditory Modelling at RSRE*, volume 2, 1987.
- 788 C.J. Plack, A.J. Oxenham, A.M. Simonson, C.G. O’Hanlon, V. Drga, and D. Ar-  
789 ifianto. Estimates of compression at low and high frequencies using masking  
790 additivity in normal and impaired ears. *J. Acoust. Soc. Am.*, 123(6):4321–4330,  
791 2008.
- 792 A.J. Power, J.J. Foxe, E.J. Forde, R.B. Reilly, and E.C. Lalor. At what time is  
793 the cocktail party? a late locus of selective attention to natural speech. *Eur. J.  
794 Neurosci.*, 35(9):1497–1503, 2012. doi: 10.1111/j.1460-9568.2012.08060.x.
- 795 B. Ross, S.A. Hillyard, and T.W. Picton. Temporal dynamics of selective attention  
796 during dichotic listening. *Cereb. Cortex*, 20(6):1360–71, 2010. doi: 10.1093/cer-  
797 cor/bhp201.
- 798 S. Van Eyndhoven, T. Francart, and A. Bertrand. EEG-informed attended speaker  
799 extraction from recorded speech mixtures with application in neuro-steered hear-  
800 ing prostheses. *IEEE Trans. Biomed. Eng.*, 64(5):1045–1056, 2017.
- 801 B.D. Van Veen, W. van Drongelen, M. Yuchtman, and A. Suzuki. Localization of  
802 brain electrical activity via linearly constrained minimum variance spatial filter-  
803 ing. *IEEE Trans. Biomed. Eng.*, 44(9):867–880, 1997.
- 804 D.I. Warton and F.K.C. Hui. The arcsine is asinine: the analysis of proportions in  
805 ecology. *Ecology*, 92(1):3–10, 2011.
- 806 A. Widmann, E. Schröger, and B. Maess. Digital filter design for electrophysio-  
807 logical data—a practical approach. *J. Neurosci. Methods*, 250:34–46, 2015. doi:  
808 10.1016/j.jneumeth.2014.08.002.
- 809 J. Wolpaw and H. Ramoser. EEG-based communication: improved accuracy by  
810 response verification. *IEEE Trans. Rehabil. Eng.*, 6(3):326–33, 1998.
- 811 D.D.E. Wong and K.A. Gordon. Beamformer suppression of cochlear implant arti-  
812 facts in an electroencephalography dataset. *IEEE Trans. Biomed. Eng.*, 56(12):  
813 2851–7, 2009.



- 814 D.D.E Wong, S.A. Fuglsang, J. Hjortkjær, E. Ceolini, M. Slaney, and  
815 A. de Cheveigné. A comparison of temporal response function estimation methods  
816 for auditory attention decoding. *BioRxiv*, 2018. doi: 10.1101/281345.
- 817 M. Wronkiewicz, E. Larson, and A.K. Lee. Incorporating modern neuroscience  
818 findings to improve brain-computer interfaces: Tracking auditory attention. *J.*  
819 *Neural Eng.*, 13(5):056017, 2016. doi: 10.1088/1741-2560/13/5/056017.
- 820 R. Zink, S. Proesmans, A. Bertrand, S. Van Huffel, and M. De Vos. Online detec-  
821 tion of auditory attention with mobile eeg: closing the loop with neurofeedback.  
822 *BioRxiv*, 2017. doi: 10.1101/218727.