Eelbrain: A Python toolkit for time-continuous analysis with temporal response functions

Christian Brodbeck[1*], Proloy Das[2], Joshua P. Kulasingham[3], Shohini Bhattasali[3], Phoebe Gaston[1], Philip Resnik[3] & Jonathan Z. Simon[3]

1) University of Connecticut
2) Massachusetts General Hospital
3) University of Maryland, College Park

* christianbrodbeck@me.com

Abstract

Even though human experience unfolds continuously in time, it is not strictly linear; instead, it entails cascading processes building hierarchical cognitive structures. For instance, during speech perception, humans transform a continuously varying acoustic signal into phonemes, words, and meaning, and these levels all have different, interdependent temporal structures. *Deconvolution analysis* has recently emerged as a promising tool for disentangling electrophysiological brain responses related to such complex models of perception. Here we introduce the Eelbrain Python toolkit, which makes this kind of analysis easy and accessible. We demonstrate its use, using continuous speech as a sample paradigm, with a freely available EEG dataset of audiobook listening. A companion GitHub repository provides the complete source code for the analysis, from raw data to group level statistics. More generally, we advocate a hypothesis-driven approach in which the experimenter specifies a hierarchy of time-continuous representations that are hypothesized to have contributed to brain responses, and uses those as predictor variables for the electrophysiological signal. This is analogous to a multiple regression problem, but with the addition of the time dimension. The deconvolution analysis decomposes the brain signal into distinct responses associated with the different predictor variables by estimating a multivariate temporal response function (mTRF), quantifying the influence of each predictor on brain responses as a function of time(-lags). This allows asking two questions about the predictor variables: 1) Is there a significant neural representation corresponding to this predictor variable? And if so, 2) what are the temporal characteristics of the neural response associated with it? Thus, different predictor variables can be systematically combined and evaluated to jointly model neural processing at multiple hierarchical levels. We discuss applications of this approach, including the potential for linking algorithmic/representational theories at different cognitive levels to brain responses through appropriate linking models.

Introduction

This paper introduces Eelbrain, a Python toolkit that makes it straightforward to express cognitive hypotheses as predictive computational models and evaluate those predictions against electrophysiological brain responses. The toolkit is based on the idea of decomposing brain signals into distinct responses associated with different predictor variables by estimating a multivariate temporal response function (mTRF), which maps those predictors to brain responses elicited by time-continuous stimulation (Theunissen et al., 2001; Lalor et al., 2006; David et al., 2007). This form of analysis has yielded valuable insights into the way that perception and cognitive processes unfold over time (e.g. Ding and Simon, 2012; Di Liberto et al., 2015; Broderick et al., 2018; Brodbeck et al., 2018a; Daube et al., 2019; Di Liberto et al., 2020; Brodbeck et al., 2020).

How to read this Paper

Deconvolution is a mathematical method for analyzing the relationship between a time-varying stimulus, like speech, and a time-varying response, like a measurement of brain activity. The goal of this paper is to introduce several categories of cognitive neuroscience questions that can be asked using deconvolution, and provide recipes for answering them. As such, the paper is not necessarily meant to be read in a linear fashion. The Introduction provides a general motivation for the approach and explains the underlying concepts in an accessible way. The Methods section explains the technical details and implementation in Eelbrain. The Results section demonstrates how the technique can be applied to answer specific questions. The Discussion section highlights some more advanced considerations and caveats that should be kept in mind. The accompanying GitHub repository (https://github.com/Eelbrain/Alice) provides the source code for everything discussed in the paper. In addition, the Eelbrain examples section provides many source code examples for more basic tasks.

Depending on the background of the reader, these resources can be approached differently – for example, we recommend reading the Introduction and Results sections first to get an idea of the questions that can be answered, and then referring to the Methods for more detailed background information.

Note on version 1

This manuscript and the accompanying source code are provided as a preliminary version. We encourage readers to let us know about any aspects that are left unclear or that could be otherwise improved to increase the didactic value. A good place for reporting issues or asking questions are the GitHub Issues and Discussions pages.

The convolution model for brain responses

The deconvolution approach is built on the assumption that the brain response is a time-continuous function of the stimulus (Lalor et al., 2006). Brain responses do not directly mirror a stimulus, but rather reflect a variety of transformations of that stimulus. For example, while speech is transmitted through air pressure variations in the kHz range, this signal is transformed by the auditory periphery, and macroscopic cortical responses are better described as responses to the slowly varying envelope of the original broadband signal. Thus, instead of directly predicting brain responses from the stimulus, the experimenter commonly selects one or several appropriate predictor variables to *represent* the stimulus, for example the low frequency speech envelope (Lalor and Foxe, 2010).

The convolution model is a formal specification of how the stimulus, as characterized by the predictor variables, leads to the response. The stimulus-response relationship is modeled as a linear convolution in time, as illustrated in Figure 1. A convolution kernel, or impulse response, characterizes the influence of each elementary unit in the predictor on the response. This kernel is also called the temporal response function (TRF), to distinguish it from the measured response to the stimulus as a whole. In addition to modeling an individual predictor variable (Figure 1-B,C), the convolution model can also incorporate multiple predictor variables through the assumption that responses are additive (Figure 1-D). Each predictor variable is associated with its own TRF, and thus predicts a separable response component. The ultimate response is the sum of those response components at each time point. This additive model is consistent with the fact that macroscopic measurements of electrical brain signals reflect an additive superposition of signals from different brain regions (Nunez and Srinivasan, 2006). When multiple predictor variables are jointly predicting a response, the combination of their TRFs is called a multivariate TRF (mTRF). As such, predictor variables can be thought of as hypotheses about how stimuli are represented by the brain, and multiple concurrent predictors can embody distinct hypotheses about how the stimulus is transformed across different brain regions (Brodbeck and Simon, 2020). The additive nature of the convolution model allows it to be applied to comparatively natural stimulus conditions, such as audiobook listening (Hamilton and Huth, 2020; Alday, 2019), while modeling natural variability through different predictor variables rather than minimizing it through experimental design.

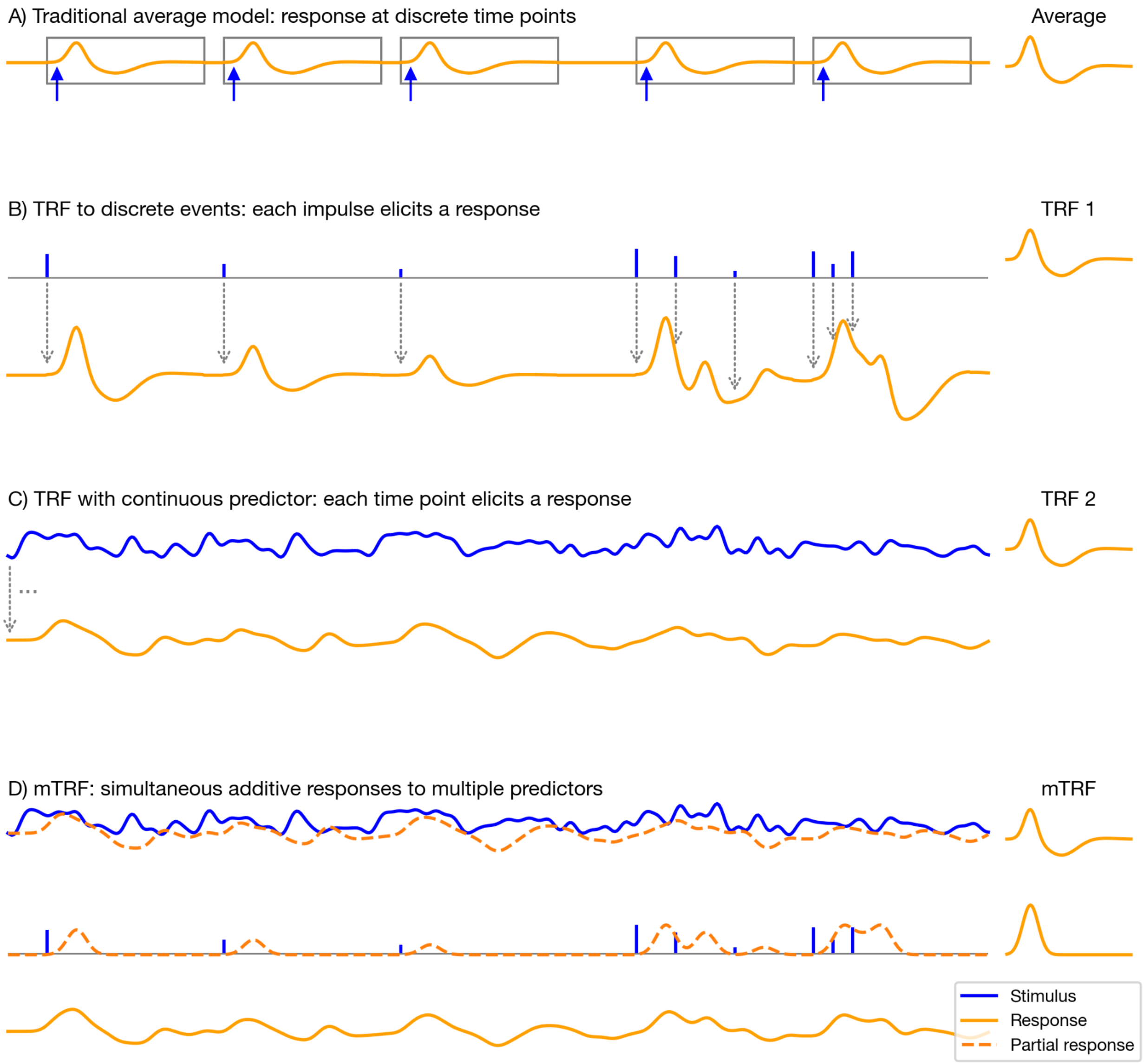
A) Traditional average model: response at discrete time points
Average
B) TRF to discrete events: each impulse elicits a response
TRF 1
C) TRF with continuous predictor: each time point elicits a response
...
TRF 2
D) mTRF: simultaneous additive responses to multiple predictors
mTRF
Stimulus
Response
Partial response

Figure 1. The convolution model for brain responses. Source code: figures/convolution.py
(A) The traditional event-related analysis method assumes that each stimulus (blue arrows) evokes an identical response, and this response can be retrieved by averaging the corresponding data segments. This model assumes that responses are separated in time.
(B) In contrast, the convolution model assumes that each time point in the stimulus could potentially evoke a response. This is implemented with time-continuous predictor variables, illustrated here with a predictor variable containing several impulses. These impulses represent events in the stimulus that are associated with a response, for example the occurrence of words. The size of each impulse determines the magnitude of the response, which could be used, for example, to allow responses to increase in magnitude the more surprising a word is. This predictor time series is convolved with some kernel characterizing the general shape of responses to this event type – the *temporal response function* (TRF), depicted on the right. Gray arrows illustrate the convolution, with each impulse producing a TRF-shaped response. As can be seen, the size of the impulse determines the magnitude of the response; this allows testing hypotheses where stimulus events differ in the magnitude of response they elicit. A major advantage over the traditional averaging model is that responses can be overlapping in time.
(C) Rather than discrete impulses, the predictor variable in this example is a continuously varying time series. Such continuously varying predictor variables can represent time-continuous properties of sensory input, for example the acoustic envelope of the speech signal. The response is dependent on the stimulus in the same manner as in (B), but now every time point of the stimulus evokes its own response shaped like the TRF and scaled by the magnitude of the predictor. Responses are therefore heavily overlapping.
(D) The multivariate TRF (mTRF) model is a generalization of the TRF model with multiple predictors: like in a multiple regression model, each predictor variable is convolved with its own corresponding TRF, resulting in multiple partial responses. These partial responses are summed to generate the actual complete response.

In most practical data analysis scenarios, the true TRFs are unknown, but the stimulus and the brain responses are known. A deconvolution algorithm addresses this, by estimating the mTRF that is optimal to predict the brain response from the predictor variables representing the stimulus. Figure 2 illustrates this with EEG responses being predicted from the speech envelope. Typically, this is a very high-dimensional problem – including several predictor variables, each of which can influence the brain response at a range of latencies. Due to the large number of parameters, mTRFs are prone to overfitting, meaning that the mTRFs learn properties of the noise in the specific dataset rather than the underlying, generalizable responses. Deconvolution methods deal with this problem by employing different regularization schemes, i.e., by bringing additional assumptions to the problem that are designed to limit overfitting (see Sparsity prior below). A further step to avoid spurious results due to overfitting is evaluating model quality with cross-validation, i.e., evaluating the model on data that was never seen during training. This step allows evaluating whether the mTRF model can generalize to unseen data and *predict* novel responses, as opposed to merely *explaining* the responses it was trained on.

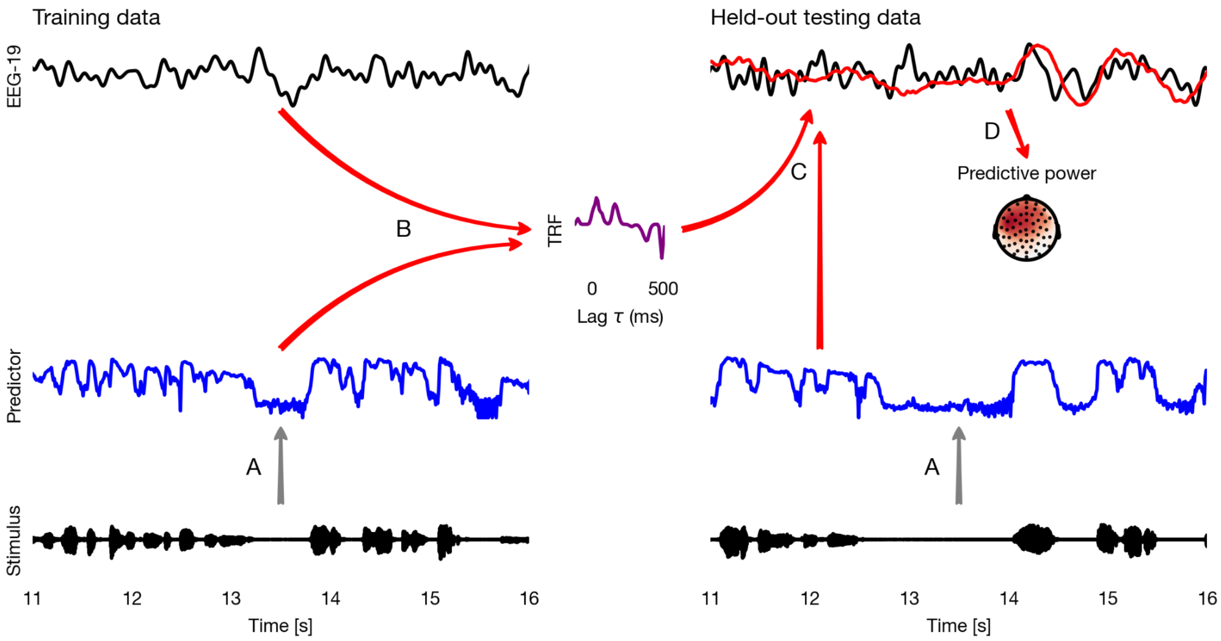

Figure 2. Deconvolution analysis of EEG speech tracking. The left half illustrates the estimation of a TRF model, the right half the evaluation of this model with cross-validation. First, the stimulus is used to generate a predictor variable, here the acoustic envelope (A). The predictor and corresponding EEG data (here only one sensor is shown) are then used to estimate a TRF (B). This TRF is then convolved with the predictor for the held-out testing data to predict the neural response to the testing data (C; measured: black; predicted: red). This predicted response is compared with the actual, measured EEG response to evaluate the predictive power of the model (D). A topographic map shows the Pearson correlation between predicted and measured EEG, estimated independently at each sensor. This head-map illustrates how the predictive power of a predictor differs across the scalp, depending on which neural sources a specific site is sensitive to. Source code: figures/Deconvolution.py.

Nonlinear responses

Convolution can only model linear responses to a given input, whereas true neural responses are known to be nonlinear. Indeed, nonlinear transformations are arguably the most interesting, because they can show how the brain transforms and abstracts away from the input, rather than merely mirroring it. We advocate a model-driven approach to study such nonlinear responses. A non-linear response can be modeled by generating a predictor variable that applies a non-linear transformation to the original input, and then predicting brain responses as a linear response to this new predictor variable. For instance, it is known that the auditory cortex is disproportionately sensitive to acoustic onsets. This sensitivity has been described with a neural model of auditory edge detection, implemented as a signal processing routine (Fishbach et al., 2001). When this edge detection model is applied to the acoustic spectrogram, this results in a spectrogram of acoustic onsets, effectively mapping regions in the signal to which specific neuron populations should respond. This transformed spectrogram as a predictor variable thus operationalizes the hypothesis that neurons perform this non-linear transformation. Indeed, such acoustic onset spectrograms are highly significant predictors of auditory MEG responses (Daube et al., 2019; Brodbeck et al., 2020). Because mTRF models can only use linear transformations of the predictor variables to predict brain responses, a significant contribution from this predictor variable suggests that the non-linear transformation captures something about the neural processes giving rise to the brain responses.

This logic for studying nonlinear responses is taken to an even further level of abstraction when language models are used to predict brain responses. For instance, linguistic theory suggests that during speech comprehension, the continuous acoustic signal is transformed into discrete representations such as phonemes and words. However, we do not yet have an explicit, computational model of this transformation that could be used to generate an appropriate predictor. Instead, experimenters can estimate the result of an implicitly specified transformation based on extraneous knowledge, such as linguistic labels and corpus data. For example, responses to phonemes or phonetic features have been modeled through predictors reflecting discrete categories (Di Liberto et al., 2015). Furthermore, a series of such investigations suggests that brain responses to speech reflect linguistic representations at different hierarchical levels (Brodbeck et al., 2018a; Weissbart et al., 2020; Gillis et al., 2021; Brodbeck et al., 2021). Such linguistic properties are commonly modeled as impulses corresponding to the onsets of words or phonemes. This does not necessarily entail the hypothesis that responses occur *at word onsets*. Rather, since the mTRFs allow responses at various latencies relative to the stimulus, such predictor variables can predict any responses that occur in an approximately fixed temporal relationship with the stimulus (within a pre-specified latency window).

During all this, it is important to keep in mind that even predictor variables that implement highly non-linear transformations are still likely to be correlated with the original stimulus input (or linear transformations of it). For example, words and phonemes are associated with specific spectro-temporal acoustic patterns which systematically relate to their linguistic significance. Before drawing conclusions about non-linear transformations implemented by the brain, it is thus always important to control for more basic input representations. In the domain of speech processing this includes at least an acoustic spectrogram and an acoustic onsets spectrogram (see Figure 3 below). The latter in particular has been found to account for responses that might otherwise be attributed to phonetic feature representations (Daube et al., 2019).

This Tutorial

For this tutorial, we use the openly available Alice dataset (Bhattasali et al., 2020) which contains data from 33 participants who listened to the first chapter of *Alice in Wonderland* (12.4 minutes; 2,129 words). The original study analyzed EEG responses to words in the story using the classical evoked response paradigm (Brennan and Hale, 2019): the EEG signals were epoched into data segments time-locked to word onsets, and then analyzed using a mass-univariate linear regression approach, regressing EEG signals to word properties. Consequently, variability due to time-varying acoustic signal properties could not be distinguished from the linguistic variables. Here we demonstrate how mTRF analysis can be used to separate acoustic and linguistic processing using the Eelbrain toolkit.

Eelbrain implements multivariate deconvolution using boosting (David et al., 2007), as well as a variety of statistical tests, and ways to extract results for further analysis with other tools. The overall implementation of Eelbrain has been guided by the goal to facilitate multivariate deconvolution, group level analysis, and visualization of results, for a general audience. The choice of boosting for deconvolution is particularly significant as it encourages TRFs with a small number of non-zero coefficients. This makes it possible to estimate mTRFs for models consisting of structured, highly correlated, and possibly redundant predictor variables (David and Shamma, 2013), as they typically occur in cognitive neuroscience deconvolutional problems.

Methods

This section describes each step towards a group level analysis, starting from the data that is included in the open Alice EEG dataset (Bhattasali et al., 2020): EEG recordings, stimulus Wave files and a comma-separated values (CSV) table with information about the words contained in the audiobook stimuli.

Time series data

In the mTRF paradigm, a time-series (for example, voltage at an EEG channel) is modeled as a linear function of one or several other time series (for example, the acoustic envelope of speech). The first step for an mTRF model is thus bringing different time-dependent variables into a common representational format. This is illustrated in Figure 3, which shows an excerpt from the first stimulus in the Alice dataset aligned to the EEG data from the first subject, along with different representations of the stimulus, which can model different neural representations.

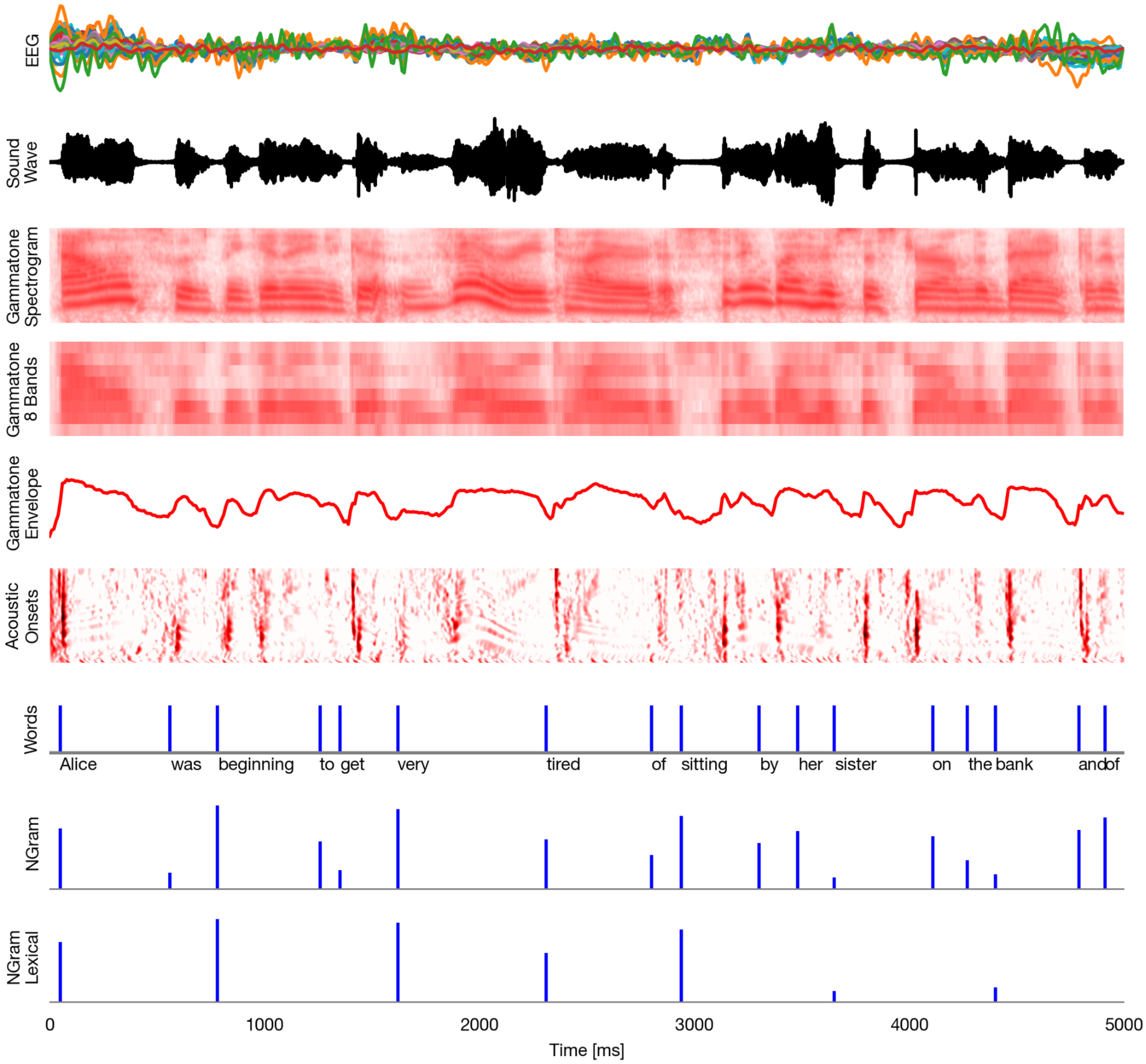
EEG
Sound Wave
Gammatone Spectrogram
Gammatone 8 Bands
Gammatone Envelope
Acoustic Onsets
Words
Alice
was
beginning
to
get
very
tired
of
sitting
by
her
sister
on
the
bank
and
of
NGram
NGram Lexical
0
1000
2000
3000
4000
5000
Time [ms]

Figure 3. Time series representations of different commonly used speech representations aligned with the EEG data. EEG: band-pass filtered EEG responses. Wave: acoustic wave-form, time-aligned to the EEG data. Gammatone: spectrogram representation modeling processing at the auditory periphery. 8 Bands: that gammatone spectrogram binned into 8 equal-width frequency bins for computational efficiency. Envelope: sum of gammatone spectrogram across all frequency bands, reflecting the broadband acoustic signal. Onsets: acoustic onset spectrogram, a transformation of the gammatone spectrogram using a neurally inspired model of auditory edge detection. Words: Uniform impulse predictor at all word onsets, predicting a constant response to all words. NGram: Impulses at word onset, scaled with word-level surprisal, estimated from an N-Gram language model, predicting brain responses to words that scale with how surprising each word is in its context. NGram/Lexical: N-Gram surprisal only at content words, predicting a response that scales with surprisal and occurs at content words only. Source code: figures/Time-series.py

As Figure 3 suggests, time series will often have different dimensions. For example, EEG data might be 2-dimensional with a time and a sensor dimension; a predictor might be one-dimensional, such as the acoustic envelope, or also have multiple dimensions, such as a spectrogram with time and frequency dimensions. To simplify working with different arbitrary dimensions, Eelbrain uses the `NDVar` (*n*-dimensional variable) class. An `NDVar` instance associates an n-dimensional numpy array (Harris et al., 2020) with *n* dimension descriptors. For example, the EEG measurements can be represented by a 2-dimensional `NDVar` with its two dimensions being `Sensor` and `UTS` dimension descriptors, characterizing the EEG sensor layout, and the time axis as a uniform time series (UTS).

The first step for the mTRF analysis is thus to import the EEG measurements and predictor variables as `NDVar` objects, and align them on the time, i.e. `UTS` dimension. The `eelbrain.load` module provides functions for importing different formats directly, such as MNE-Python objects and Wave files (but `NDVar` objects can also be constructed directly from `numpy` arrays). Keeping information about dimensions on the `NDVar` objects allows for concise code for tasks such as aligning, plotting etc. The code for Figure 2 includes an example of loading a Wave file and aligning its time axis to the EEG data through filtering and resampling.

EEG data

EEG data should be pre-processed according to common standards. In the Python ecosystem, MNE-Python offers a unified interface to a variety of EEG file formats and preprocessing routines (Gramfort et al., 2014). Here, we rely on the data preprocessed with independent component analysis (ICA) provided by the Alice EEG dataset (see Bhattasali et al., 2020). However, a crucial additional step is filtering and downsampling the raw data. When analyzing continuous electrophysiological recordings, removing low frequencies (i.e., high-pass filtering) takes the place of baseline correction, by removing high-amplitude slow drifts in the EEG signal which would otherwise overshadow effects of interest. Removing high frequencies, beyond the signals of interest (low-pass filtering), reduces noise and, more crucially, allows conducting the analysis at a lower sampling rate. This makes the analysis faster, because deconvolution is computationally demanding, and processing times scale with the sampling rate. The major cortical phase-locked responses decrease quickly above 10 Hz (Ding et al., 2014), although there can be exceptions, such as a pitch-following response up to 80 Hz (Kulasingham et al., 2020). For the purpose of this tutorial we are interested in the range of common ERP components and apply a 0.5-20 Hz band-pass filter. Theoretically, a sampling rate 2.5 times the highest frequency is sufficient for a faithful representation of the signal (Nunez and Srinivasan, 2006), but a higher rate results in smoother results which can be desirable for secondary analysis and visualization. Here we conduct the analysis with a sampling rate of 100 Hz.

EEG data usually contains markers that indicate the start of the stimulus presentation. These can be used to quickly extract EEG data time-locked to the stimuli in the required time series format, i.e. a 2-dimensional `NDVar` with `Sensor` and `UTS` dimensions (see Figure 2 code).

Predictor variables

Any hypothesis about time-locked neural processing that can be quantified can be represented by a time series derived from the stimulus. Here we will illustrate two approaches: The first approach implements hypotheses about spectro-temporal transformations of the acoustic signal by directly applying those transformations to the speech waveform. The second approach implements hypotheses about linguistic processing based on experimenter-determined, discrete linguistic events.

Time-continuous predictor variables: gammatone spectrogram and derivatives

A common starting point for modeling acoustic responses is through a model of the cochlear transformation of the sound. Here we use the gammatone spectrogram method to estimate cochlear transformations (Patterson et al., 1992; Heeris, 2018). A gammatone-spectrogram is initially a high-dimensional representation, with more than a hundred time series representing the acoustic power at different frequency bands. For computational efficacy, we reduce the number of bands by summarizing several contiguous bands into one. Here we use 8 bands as a compromise that leaves the global acoustic structure intact (Figure 3). An extreme form of this dimension reduction is using the acoustic envelope, which summarizes the entire spectrogram with a single band.

In addition to representing raw acoustic features, the auditory cortex is known to prominently represent acoustic onsets (Daube et al., 2019). Here we model such representations by applying the neurally inspired auditory edge detection transformation to the gammatone spectrogram (Brodbeck et al., 2020). It is also common to approximate such a transformation through the half-wave rectified derivative of the acoustic envelope (Fiedler et al., 2017; Daube et al., 2019).

Because these predictor variables will be used repeatedly, it is convenient to generate them once and save them to disk. The script `predictors/make_gammatone.py` loops through all stimuli and computes a high-dimensional gammatone spectrogram, a 2-dimensional `NDVar` with `frequency` and `UTS` dimensions, and saves them as Python pickle files via the `eelbrain.save` module. The script `predictors/make_gammatone_predictors.py` loads these high-dimensional spectrograms via the `eelbrain.load` module and resamples them to serve as predictors, and it also applies the onset transformation and saves the resulting predictors.

Discrete predictor variables

The analysis of linguistic representations, which cannot be derived directly from the sound files, commonly relies on forced alignment, a method that infers time-stamps for phoneme and word boundaries by comparing the sound files with a transcript. An example of an open source forced aligner with extensive documentation is the Montreal Forced Aligner (e.g. McAuliffe et al., 2017). Because the forced alignment procedure requires some additional steps that are well documented by the respective aligners we skip it here, and instead use the word-onset time-stamps provided with the Alice dataset.

Discrete predictors come in two varieties: constant magnitude impulses and variable magnitude impulses (see Figure 3, lower half). Constant magnitude impulses always have the same magnitude, for example an impulse of magnitude one at each word onset. Such a predictor implements the hypothesis that all words are associated with a shared characteristic brain response, similar to an ERP. The deconvolution algorithm will then determine the latencies relative to those impulses at which the brain exhibits a consistent response. Variable magnitude impulses implement the hypotheses that the brain response to each word varies systematically with some quantity. For example, the N400 is assumed to co-vary with how surprising the word is in its context. A predictor with an impulse at each word onset, whose magnitude is equal to the surprisal of that word, will enable predicting a stereotyped response that varies in amplitude depending on the given variable. The linking hypothesis here is that for each event, the brain responds with population activity that scales in amplitude with the magnitude of the predictor.

The Alice dataset comes with a table including all time-stamps and several linguistic properties (`stimuli/AliceChapterOne-EEG.csv`). Each row of this table represents one word, and contains the time point at which the word starts in the audio file, as well the surprisal values that were used to relate the EEG signal to several language models in the original analysis (Brennan and Hale, 2019). Such a table listing event times and corresponding feature values is sufficient for constructing appropriate regressors on the fly, and has a much lower memory footprint than a complete time-series. The script `predictors/make_word_predictors.py` converts the table into the Eelbrain `Dataset` format that can be directly used to construct the regressor as an `NDVar`. To keep a common file structure scheme with the continuous regressors, such a table is generated for each stimulus.

Deconvolution

Background: The convolution model

The assumption behind the mTRF approach is that the dependent variable is the result of a convolution (linear filtering) of one or several predictor variables with corresponding response functions. For a single predictor variable, the model is formulated as the convolution of the predictor variable with a one-dimensional filter kernel:

$$y_t = \sum_{\tau=\tau_{min}}^{\tau_{max}} h_\tau x_{t-\tau}$$

Here h represents the filter kernel, also known as TRF, and τ enumerates the delays between y and x at which x can influence y. To extend this approach to multiple predictor variables, it is assumed that the individual filter responses are additive. In that case, x consists of n predictor time series and thus has 2 dimensions, one being the time axis and the other consisting of n different predictor variables. The corresponding multivariate TRF (mTRF) is also 2-dimensional, consisting of one TRF for each predictor variable:

$$y_t = \sum_{i=0}^{n} \sum_{\tau=\tau_{min}}^{\tau_{max}} h_{i,\tau} x_{i,t-\tau}$$

This model allows predicting a dependent variable y, given predictors x and mTRF h.

In neural data analysis scenarios, typically the measured brain response and the stimuli are known, whereas the filter kernel h is unknown. This leads to two reformulations of the general deconvolution problem in which x and y are known, and h is to be estimated; these represent alternative approaches for deconvolution analysis. In the first, the so-called forward or encoding model, h is optimized to predict brain responses from stimulus representations. In the second, the so-called backward, or decoding model, h is optimized to reconstruct a stimulus representation from the neural measurements. The problems can both be expressed in the same general form and solved with the same algorithms. By way of example, we here use the boosting algorithm implemented in Eelbrain (see Background: Boosting implementation below).

Deconvolution: Forward model (encoding)

Given a continuous measurement and one or more temporally aligned predictor variables, the reverse correlation problem consists of finding the filter kernel, or mTRF that optimally predicts the response from the stimuli. The result of the convolution now becomes the predicted response

$$\hat{y}_t = \sum_{i=0}^{n} \sum_{\tau=\tau_{min}}^{\tau_{max}} h_{i,\tau} x_{i,t-\tau} \tag{1}$$

The goal of the algorithm estimating the mTRF h is to minimize the difference between the measured response y_t and the predicted response $\hat{y}_t$. Figure 2 illustrates estimation of a forward model for EEG data.

The `eelbrain.boosting` function provides a high level interface for estimating mTRFs and returns a `BoostingResult` object with different attributes containing the mTRF and several model fit metrics for further analysis. Usually, for a forward model, the brain response is predicted from the predictors using positive lags. For example,

```
trf = boosting(eeg, envelope, 0, 0.500)
```

would estimate a TRF to predict EEG data from the acoustic envelope, with lags ranging from 0 to 500 ms (Eelbrain uses seconds as the default time unit). This means that an event at a specific time in the acoustic envelope could influence the EEG response in a window between 0 and 500 ms later. If the dependent variable has multiple measurements, for example as here multiple EEG channels, Eelbrain automatically assumes a mass-univariate approach and estimates a TRF for each channel. Negative lags, as in

```
trf = boosting(eeg, envelope, -0.100, 0.500)
```

are non-causal in the sense that they assume a brain response that precedes the stimulus event. Such estimates can nevertheless be useful for at least two reasons. First, if the stimulus genuinely represents information in time, then non-causal lags can be used as an estimate of the noise floor. As such they are analogous to the baseline in ERP analyses, i.e., they indicate how variable TRFs are at time points at which no true response is expected. Second, when predictor variables are experimenter-determined, the temporal precision of the predictor time series might often be reduced, and information in the acoustic speech signal might in fact precede the predictor variable. In such cases, the negative lags might be diagnostic. For example, force-aligned word-and phoneme onsets assume the existence of strict boundaries in the speech signal, when in fact the speech signal can be highly predictive of future phonemes due to coarticulation (e.g. Salverda et al., 2003).

An advantage of the forward model is that it can combine information from multiple predictor variables. The boosting algorithm in particular is robust with a large number of (possibly correlated) predictor variables (e.g., David and Shamma, 2013). Eelbrain supports two ways to specify multiple predictor variables. The first is using a multi-dimensional predictor, for example a two-dimensional `NDVar`, `spectrogram` with `frequency` and `UTS` dimensions. These are used with the boosting function just like one-dimensional time series, and will be treated as multi-dimensional predictor variables, i.e., they will be jointly used for the optimal prediction of the dependent variable:

```
mtrf = boosting(eeg, spectrogram, 0, 0.500)
```

The second option for using multiple predictor variables is specifying them as a list of `NDVar` (one and/or two dimensional), for example:

```
mtrf = boosting(eeg, [envelope, spectrogram], 0, 0.500)
```

Deconvolution: Backward model (decoding)

Instead of predicting the EEG measurement from the stimulus, the same algorithm can attempt to reconstruct the stimulus from the EEG response. The filter kernel is then called a decoder. This can be expressed with (1), but now y_t refers to a stimulus variable, for example the speech envelope, and $x_{i,t}$ refers to the EEG measurement at sensor i and time t. Accordingly, a backward model can be estimated with the same boosting function. For example,

```
decoder = boosting(envelope, eeg, -0.500, 0)
```

estimates a decoder model to reconstruct the envelope from the EEG data. Note the specification of delay values τ in the boosting function, from the point of view of the predictor variable: because each point in the EEG response reflects the stimulus preceding it, the delay values are negative, i.e., a given point in the EEG response should be used to reconstruct the envelope in the 500 ms window preceding the EEG response.

Because the EEG channels now function as the multivariate predictor variable, all EEG channels are used jointly to reconstruct the envelope. An advantage of the backward model is thus that it combines data from all EEG sensors to reconstruct a stimulus feature. It thus provides a powerful measure of how much information about the stimulus is contained in the brain responses, taken as a whole. A downside is that it does not provide a straight-forward way for distinguishing responses that are due to several, correlated predictor variables. For this reason, we will not further discuss it here.

Background: The boosting algorithm

The deconvolution problem, i.e., finding the optimal filter kernels in forward and backward models, can be solved with different approaches. The Eelbrain toolkit implements the boosting algorithm which is resilient to over-fitting and performs well with correlated predictor variables (David et al., 2007; David and Shamma, 2013). For an alternative approach with regularized regression see (Crosse et al., 2016); for an approach that combines deconvolution with source localization as a unified problem see (Das et al., 2020).

Boosting starts by dividing the data into training and validation folds, and assuming an empty mTRF (all values $h_{i,\tau}$ set to 0). It then repeatedly uses the training data to find the element in the filter kernel which, when changed by a fixed small amount `delta`, leads to the largest error reduction. For multiple predictors, the search is performed over all the predictors as well as time lags, essentially letting the different predictors compete to explain the dependent variable. After each such `delta` change, the validation data is consulted to verify that the error is also reduced in the validation data. Once the error starts increasing in the validation set, the training stops. This early stopping strategy prevents the model from overfitting to the training data.

Because the default implementation of the boosting algorithm constructs the kernel from impulses, this can lead to temporally sparse TRFs. In order to derive smoother TRFs, TRFs can be constructed from a basis of smooth windows instead. In Eelbrain, the basis shape and window size are controlled by the `basis` and `basis_window` parameters in the `boosting` function.

When using multiple predictors, it might be undesirable to stop the entire model training when a single predictor starts overfitting. In that case, the `selective_stopping` option allows freezing only the predictor which caused the overfitting, while the TRF components corresponding to the remaining predictors are trained, until all predictors are frozen.

Finally, the default error metric for evaluating model quality is the widely used ℓ_2 error. Due to squaring, the ℓ_2 error is disproportionately sensitive to time points with large errors. Specifically in electrophysiology, large errors are typically produced by artifacts, and it is undesirable to give such artifacts a large influence on the model estimation. Since it is not trivial to exclude time-intervals containing such artifacts from the analysis in continuous data, Eelbrain also implements the ℓ_1 error norm through the `error='l1'` argument, which improves robustness against such outlier data.

Cross-validation

By default, the boosting function trains an mTRF model on all available data – cross-validation is enabled by setting the `test` parameter to `test=True`. Since the boosting algorithm already divides the data into training and validation sets, enabling cross-validation entails splitting the data into three segments for each run: training, validation and test set. While the training and validation segments are consulted for estimating the TRFs, the test segment does not influence the generation of the TRFs at all. Only once the TRF estimation is finalized, the final TRF is used to predict the responses in the test segment. To use the data optimally, Eelbrain automatically implements k-fold cross-validation, whereby the data is divided into k partitions, and each partition serves as the test set once (see Data partitions Eelbrain example). Thus, through k-fold cross-validation, the whole response time-series can be predicted from unseen data. The proportion of the explained variability in this response constitutes an unbiased estimate of the predictive power of the given model. The `BoostingResult` object returned by the `boosting` function contains all these metrics as attributes for further analysis.

Evaluating predictive power

In practice, a research question can often be operationalized by asking whether a specific predictor variable is neurally represented, i.e., whether it is a significant predictor of brain activity. Different predictor variables of interest are very often correlated in naturalistic stimuli such as speech. It is thus important to test the explanatory power of a given variable while controlling for the effect of other, correlated variables. A powerful method for this is comparing the predictive power of minimally differing sets of predictor variables using cross-validation.

In this context, we use the term model to refer to a set of predictor variables, and model comparison refers to the practice of comparing the predictive power of two models on held-out data. It is important to re-estimate the mTRFs for each model under investigation to determine the effective predictive power of that model, because mTRFs are sensitive to correlation between predictors and can thus change depending on what other predictors are included during estimation.

When building models for a specific model comparison, we recommend a hierarchical approach: both models should include lower-level properties that the experimenter wants to control for, while the models should differ only in the feature of interest. For example, to investigate whether words are associated with a significant response after controlling for acoustic representations, one could compare the explained variability of (2) and (3):

gammatone spectrogram + onset spectrogram (2) *gammatone spectrogram + onset spectrogram + word onset impulses* (3)

If the model in (3) is able to predict held-out data better than that in (2) then this difference can be attributed to a predictive power of word onset impulses over and above the spectrogram-based representations.

Estimating such mTRF models is the computationally most demanding part of this analysis. For this reason, it usually makes sense to store the result of the individual estimates. Script `analysis/estimate_trfs.py` implements this, by looping through all subjects, fitting mTRFs for multiple models, and saving the results for future analysis.

Group analysis

In order to statistically answer questions about the predictive power of different models we will need to combine the data from different measurements, usually different subjects. To combine data from multiple subjects along with meta-information such as subject and condition labels Eelbrain provides the `Dataset` class, analogous to data-table representations in other statistics programs such as a dataframe in `R` or `Pandas`, but with the added capability of handling data from `NDVars` with arbitrary dimensionality.

A standard way of constructing a Dataset is collecting the individual cases, or rows of the desired data table and then combining them. The following short script provides a template for assembling a table of model predictive power for several subjects and two models (assuming the mTRF models have been estimated and saved accordingly):

```
cases = []
for subject in ['1', '2', '3']:
    for model in ['sgrams', 'srams+words']:
        mtrf = load.unpickle(f"path/to/{subject}_{model}.pickle")
        cases.append([subject, model, mtrf.proportion_explained])
column_names = ['subject', 'model', 'explained']
data = Dataset.from_caselist(column_names, cases)
```

Thus, even though the `proportion_explained` attribute might contain 64 EEG channels (i.e. an `NDVar` with `sensor` dimension) it can be handled as a single column in this data-table.

Statistical tests

Statistical analysis of mTRFs faces the issue of multiple comparison common in neuroimaging work (Maris and Oostenveld, 2007). One way around this is to derive a univariate outcome measure, for example, the average predictive power at a pre-specified group of sensors:

```
sensors = ['45', '34', '35']
data['explained_average'] = data['explained'].mean(sensor=sensors)
```

Eelbrain implements a limited number of basic univariate statistical tests in the `test` module, but more advanced analysis can be performed after exporting the data into other libraries. All univariate entries in a `Dataset` can be transferred to a `pandas.DataFrame` (Reback et al., 2021) with

```
dataframe = data.as_dataframe()
```

for analysis with other Python libraries like Pingouin (Vallat, 2018), or saved as text file with

```
data.save_txt('path.txt')
```

to be transferred to another statistics environment like `R` (R Core Team, 2021).

Instead of restricting the analysis to a-priori sensor groups, Eelbrain implements several mass-univariate tests (Nichols and Holmes, 2002; Maris and Oostenveld, 2007; Smith and Nichols, 2009). These tests, exposed in the `eelbrain.testnd` module (for n-dimensional tests), are generally based on calculating a univariate statistic at each outcome measure (for example, a t value corresponding to a repeated-measures t-test comparing the predictive power of two models at each EEG sensor), and then using a permutation-based approach for estimating a null-distribution for calculating p-values that control for family-wise error. In Eelbrain, these tests can be applied to `NDVars` like univariate tests, with additional arguments for controlling the precise method for correcting for multiple comparisons. The script to Figure 5 demonstrates a complete group analysis, from loading pickled mTRF models into a `Dataset` to plotting statistical results.

TRF analysis

While predictive power is the primary measure to assess the quality of a model, the TRFs themselves also provide information about the temporal relationship between brain response and the stimuli. A TRF describes an estimate of the brain's response to an impulse stimulus of magnitude 1 (i.e., the impulse response) to the corresponding predictor at different latencies, similar to an ERP to a simple stimulus. For example, the TRF estimated to an acoustic envelope representation of speech commonly resembles the ERP to simple tone stimuli. The mTRFs can thus be analyzed in ways that are analogous to ERPs, for example using mass-univariate tests or based on component latencies. Figure 4 demonstrates an analysis of TRFs and mTRFs corresponding to different auditory features.

Results

Auditory response functions

Speech is a complex acoustic signal, and brain responses are highly sensitive to different acoustic features in speech. Figure 4 gives an overview of response functions associated with acoustic features that are commonly used to predict brain responses. The acoustic features are shown in Figure 4-A: The upper panel shows the log-transformed gammatone spectrogram, quantifying acoustic energy as a function of time in different frequency bands. The gammatone filters simulate response characteristics of the peripheral auditory system. A simplified, one-dimensional representation of this spectrogram is the envelope, which is the summed energy across all frequency bands (blue line). The auditory system is additionally known to be particularly sensitive to acoustic onsets. The lower panel of Figure 4-A shows an acoustic onset spectrogram based on a neurally inspired acoustic edge detection model (Fishbach et al., 2001), as described in (Brodbeck et al., 2020), although a similar predictor can be derived from the half-wave rectified derivative of the spectrogram or enveloppe (Fiedler et al., 2017; Daube et al., 2019). Again, a simplified one-dimensional version of this predictor, summing across all bands, signifies the presence of any onsets across frequency bands (blue line).

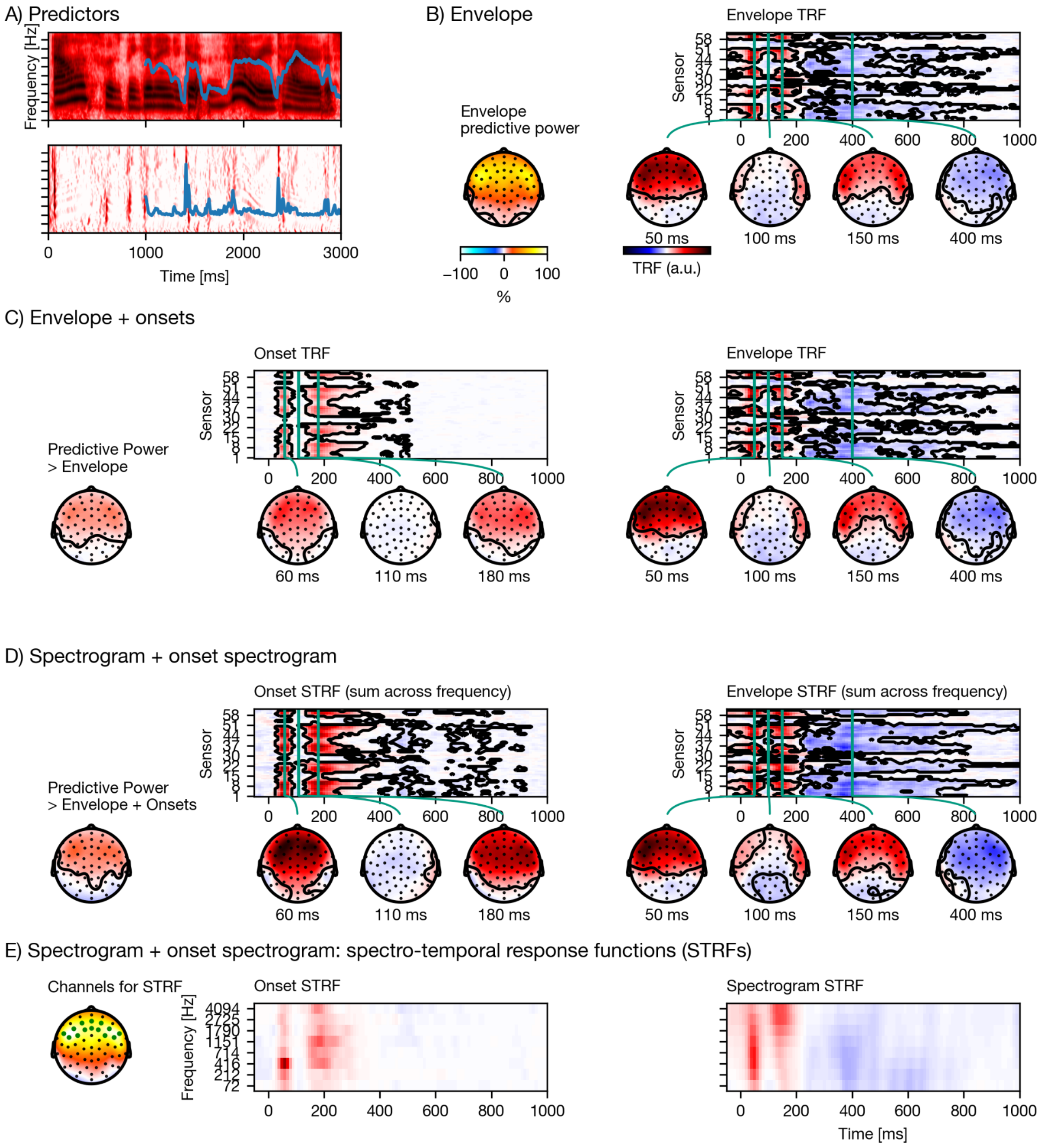
A) Predictors
Frequency [Hz]
0
1000
2000
3000
Time [ms]
B) Envelope
Envelope predictive power
−100
0
100
%
Envelope TRF
Sensor
58
51
44
37
30
22
15
8
1
0
200
400
600
800
1000
50 ms
100 ms
150 ms
400 ms
TRF (a.u.)
C) Envelope + onsets
Predictive Power > Envelope
Onset TRF
60 ms
110 ms
180 ms
Envelope TRF
50 ms
100 ms
150 ms
400 ms
D) Spectrogram + onset spectrogram
Predictive Power > Envelope + Onsets
Onset STRF (sum across frequency)
60 ms
110 ms
180 ms
Envelope STRF (sum across frequency)
50 ms
100 ms
150 ms
400 ms
E) Spectrogram + onset spectrogram: spectro-temporal response functions (STRFs)
Channels for STRF
Onset STRF
Frequency [Hz]
4094
2725
1790
1151
714
416
212
72
Spectrogram STRF
Time [ms]

Figure 4. Auditory temporal response functions.

(A) Common representations of auditory speech features: an auditory spectrogram (upper panel), and the acoustic envelope, tracking total acoustic energy across frequencies over time (blue line); and an acoustic onset spectrogram (lower panel), also with a one-dimensional version characterizing the presence of acoustic onsets over time (blue line).

(B) Brain responses predicted from the acoustic envelope of speech alone. Left: Cross-validate predictive power is highly significant ($p < .001$) at a large cluster covering most anterior sensors. The black outline marks a cluster in which the predictive power is significantly larger than chance ($p \leq .05$, family-wise error correction for the whole head map). Right: the envelope TRF – the y-axis represents the different EEG channels (in an arbitrary order), and the x-axis represents predictor-response time lags. The green vertical lines indicate specific (manually selected) time points of interest, for which head map topographies are shown. The black outlines mark significant clusters ($p \leq .05$, correction for the whole TRF). For the boosting algorithm, predictors and responses are typically normalized, and the TRF is here analyzed and displayed in this normalized scale.

(C) Results for an mTRF model including the acoustic envelope and acoustic onsets (blue lines in A). The left-most head map shows the increase in predictive power over the TRF mode using just the envelope ($p < .001$). Details are analogous to (B).

(D) Results for an mTRF model including spectrogram and onset spectrogram, further increasing predictive power over the one-dimensional envelope and onset model ($p < .001$). Since the resulting mTRFs distinguish between different frequencies in the stimulus, they are called spectro-temporal response functions (STRFs). In (D), these STRFs are visualized by summing across the different frequency bands.

(E) To visualize the sensitivity of the STRFs to the different frequency bands, STRFs are averaged across sensors that are sensitive to the acoustic features. The relevant sensors are marked in the head map on the left, which also shows the predictive power of the full spectro-temporal model. Because boosting results in sparse STRFs, especially when predictors are correlated, as are adjacent frequency bands in a spectrogram, STRFs were smoothed across frequency bands for visualization.

a.u.: arbitrary units

To illustrate auditory features of increasing complexity we analysed the following (m)TRF models:

$$\textit{acoustic-envelope} \tag{4}$$

$$\textit{acoustic-envelope + acoustic-onsets} \tag{5}$$

$$\textit{acoustic-spectrogram + onset-spectrogram} \tag{6}$$

Figure 4-B shows response characteristics to the acoustic envelope alone. A topographic head-map shows the envelope's predictive power, expressed as percentage of the predictive power of the full spectro-temporal model (6) at the best EEG sensor (Figure 4-D). The envelope alone is already a very good predictor of held-out EEG responses, reaching 74% of the full model at anterior electrodes. The TRF to the envelope shows features characteristic of auditory evoked responses to simple acoustic stimuli such as tones or isolated syllables, including a P1-N1-P2 sequence.

Figure 4-C shows the result of adding the one-dimensional acoustic onset predictor to the model. Together, onset and envelope significantly improve the prediction of held-out responses compared to just the envelope ($p < .001$), indicating that the onset representation is able to predict some aspects of the EEG responses that the envelope alone cannot. The typical TRF to the onsets is of shorter duration than the envelope, characterized by two prominent peaks around 60 and 180 ms. The envelop TRF here is not much affected by adding the onset to the model (compare with Figure 4-B).

Figure 4-D shows the additional benefit of representing the envelope and onsets in different frequency bands, i.e., predicting EEG from an auditory spectrogram and an onset spectrogram (here 8 bands were used in each, for a total of 16 predictors). As the predictors are 2-dimensional (frequency × time), the resulting mTRFs are 3-dimensional (frequency × lag × EEG sensor), posing a challenge for visualization on a two-dimensional page. One approach, based on the assumption that response functions are similar across frequency bands, is to sum responses across frequency bands. As shown in Figure 4-D, this indeed results in response functions that look very similar to the one-dimensional versions of the same predictors (Figure 4-C). To visualize how the response functions differ for different frequency bands in the spectrograms, Figure 4-E shows the full spectro-temporal response functions (STRFs), averaged across electrodes that are most sensitive to the auditory stimulus features.

In sum, while the acoustic envelope is a powerful predictor of EEG responses to speech, additional acoustic features can improve predictions further, suggesting that they characterize neural representations that are not covered by the envelope. While the benefit in prediction accuracy might seem limited, a more important property of different predictors is that they can track different neural representations. This is important because different neural representations can have different response characteristics in different situations. For example, acoustic onsets might be especially important in segregating multiple auditory streams (Brodbeck et al., 2020; Fiedler et al., 2019).

Categories of events: Function and content words

Given the powerful response to acoustic features of speech, it is important to take these responses into account when investigating linguistic representations. To illustrate the advantage of an mTRF analysis that can take into account the acoustic stimulus features, we revisit an old question: do brain responses differentiate between content and function words? In a first, naive approach, we ask literally whether brain responses differ between function and content words, while ignoring any potential confounding influences from acoustic differences. For this, we compare the predictive power of models (7) and (8), all based on predictors with equal magnitude impulses at word onsets:

All-words + function-words + content-words (7)

All-words (8)

Note that *all-words* and the two word class-based predictors are mathematically redundant because the *all-words* vector equals the sum of the *function-words* and *content-words* vectors. However, because of the sparsity prior employed through the boosting algorithm and cross-validation, such redundant predictors can actually improve a model – for example, if there is actually a shared response among all words, then learning this response once as a TRF to *all-words* is easier and sparser than learning it twice, once for *function-words* and once for *content-words* separately (see also Sparsity prior below).

The predictive power of the model with the word class distinction (7) is significantly larger compared to the model without (8) at a right anterior electrode cluster (p = .002, Figure 5-A, top). To determine why the word class distinction improves the model's predictive power, we compare the TRFs to function and content words (Figure 5-B). Both TRFs are reconstructed from the relevant model components: Whenever a function word occurs in the stimulus, there will be both an impulse in the *all-words*, and one in the *function-words* predictor (and vice-versa for content words). Accordingly, the full EEG response occurring at the function word is split into two components, one in the *all-words*, and one in the *function-words* TRF. In order to visualize these full EEG responses, the displayed TRF for function words consists of the sum of the TRFs of the *all-words* and the *function-words* predictors, and the TRF for content words consists of the sum of the TRFs of *all-words* and *content-words*. This comparison suggests that function words are associated with a larger positive response at anterior sensors than content words.

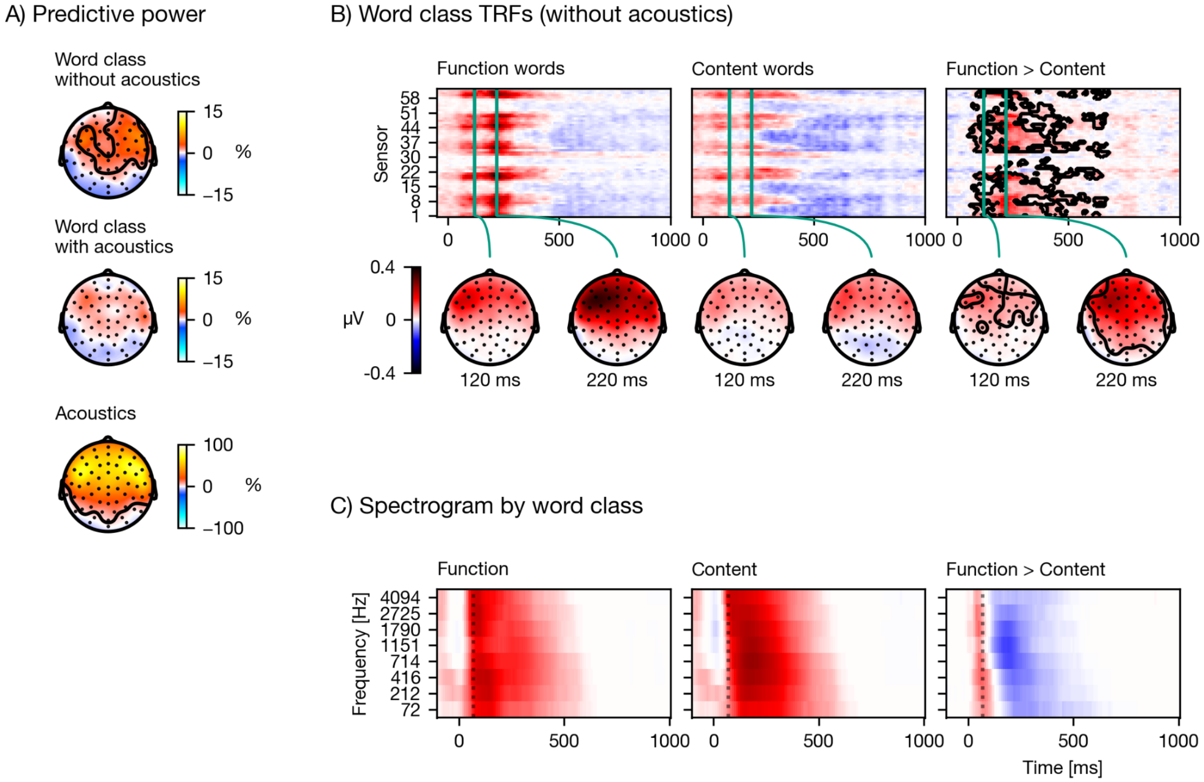

Figure 5. The difference in response to content and function words explained by acoustic differences. Source code: figures/Word-class-acoustic.py

(A) Model comparisons of predictive power across the scalp. Each plot shows a head-map of the difference in predictive power between two models. Top: (7) > (8); middle: (9) > (10); bottom: (10) > (8).

(B) Brain responses occurring after function words differ from brain responses after content words. Responses were estimated from the TRFs of model (7) by adding the word-class specific TRF to the *words-all* TRF. The outlined region is significantly different between function and content words based on a mass-univariate related-measures *t*-test.

(C) Function words are associated with a sharper acoustic onset than content words. The average spectrograms associated with function and content words were estimated with deconvolution of the acoustic spectrogram, analogous to brain responses in (B). A dotted line is plotted at 70 ms to help visual comparison. Color-scale is normalized.

Next, to ask whether this difference in responses might be due to acoustic confounds, we control for brain responses to acoustic features in both models and compare (9) with (10):

acoustic-spectrogram + onset-spectrogram + all-words + function-words + content-words (9)

acoustic-spectrogram + onset-spectrogram + all-words (10)

In contrast to the comparison without acoustics, the comparison of the predictive power of (9) with (10) does no longer indicate a significant difference (p = .071, Figure 5-A, middle). This suggests that the information in the acoustic predictors can explain the difference between models (7) and (8). To directly characterize the influence of the acoustic features we plot the predictive power of the acoustic features by comparing models (10) with (8). This comparison suggests that acoustic features are highly predictive of brain signals at anterior sensors (Figure 5-A, bottom), encompassing the region in which the word class distinction originally showed an effect.

If the difference between brain responses to function and content words disappears when controlling for acoustic features, that suggests that acoustic features should differ between function and content words. We can assess this directly with a convolution model. First, we estimate filter kernels (analogous to the mTRFs) to predict the auditory spectrogram from models (8) and (7) (script: analysis/estimate_word_acoustics.py). To statistically evaluate these results, we use 15-fold cross-validation and treat the predictive accuracy from each test fold as an independent estimate of predictive power (for a similar approach see Etard et al., 2019). With predictive power averaged across frequency bands, the model distinguishing function and content words is significantly better at predicting the spectrogram than the model treating all words equally (t_{14} = 14.10, p < .001), suggesting that function and content words indeed differ acoustically. Finally, the filter kernels to function and content words from this analysis can be interpreted as average acoustic patterns corresponding to the two word classes (Figure 5-C). This comparison suggests that function words on average have a sharper acoustic onset. Since auditory cortex responses are known to be sensitive to acoustic onsets, a reasonable explanation for the difference in neural responses by word class, when not controlling for acoustic features, is that it reflects these sharper acoustic onsets of function words.

Discussion

While the sections above provide recipes for various aspects of deconvolution analysis, we use this section to first discuss a number of advanced considerations and caveats, and possible extensions.

TRF analysis vs. predictive power

In the various Results sections, we always tested the predictive power of a predictor variable before analyzing the corresponding TRF. This has good reason: model comparisons based on predictive power are generally more conservative than TRF estimates. As in conventional regression models, if two predictors are correlated, that means that they might share some of their predictive power. Model comparisons address this by testing for the *unique* predictive power of a variable, after controlling for all other variables, i.e., by testing for variability in the dependent measure that can *only* be explained by the predictor under investigation. The mTRF parameters (i.e., regression coefficients) cannot properly be disentangled in this way (Freckleton, 2002), and will usually divide the shared explanatory power amongst themselves. This means that mTRF estimates may always be contaminated by responses to correlated variables, especially when the correlations among predictor variables are high. This consideration highlights the importance of testing predictive power of individual model components before interpreting the corresponding TRFs for avoiding spurious conclusions due to correlated predictors.

Sparsity prior

In their general formulation, mTRF analysis is just a regression analysis, albeit a high dimensional one. Such high dimensional analysis methods are almost always marred by overfitting. Presence of a large number of free parameters and correlated predictor variables makes the mTRFs prone to discover noise patterns that are eccentric to the particular dataset, which are poorly generalizable. Most regression analysis methods deal with this problem through regularization using a well-informed prior and some form of cross-validation. These regularization schemes vary considerably; some employ an explicitly formulated prior, e.g., the regularization in ridge regression (Crosse et al., 2016); while others are defined on the algorithmic level, e.g., the early stopping criterion for boosting (see Background: The boosting algorithm). The implicit prior in the boosting algorithm promotes sparsity, i.e. it forces unimportant filter kernel coefficients to exactly 0. It has been shown that for problems with large numbers of correlated predictors, the boosting algorithm might be preferable to other sparsity enforcing algorithms such as lasso (Hastie et al., 2007).

This sparsity prior has some consequences that might be counterintuitive at first. For example, in regression models it is common to center predictors. This does not affect the explanatory power of individual regressors, because a shift in the mean of one predictor will simply lead to a corresponding shift in the coefficient for the intercept term. In contrast to this, a sparsity prior will favor a model with smaller coefficients. Consequently, an uncentered predictor that can explain responses with a small coefficient in the intercept term will be preferable over a centered predictor that requires a large intercept coefficient.

Another consequence of the same preference for sparser models is that sometimes mathematically redundant predictors can improve the predictive power of a model. An example is that when splitting an impulse predictor into two categories, such as when dividing words into function and content words, the original all-words predictor does not necessarily become redundant (see Categories of events: Function and content words). This can make model construction more complex, but it can also be informative, for example by showing whether there is a common response component to all words, that can be learned better by including an *all-words* predictor, in addition to word class specific responses.

Discrete events

In practice, the assumption of millisecond-precise time-locking to force-aligned features might seem hard to defend. For example, identification of clear word- and phoneme boundaries is an artificial imposition, because the actual acoustic-phonetic features blend into each other due to co-articulation. However, the analysis only requires time-locking on average to produce consistent responses – as is the case with classical ERPs. However, one might thus want to consider alternatives, such as the words' uniqueness point instead of word onset.

Impulse is not the only coding that is available. Impulses are based on the linking hypothesis of a fixed amount of response per impulse. An alternative linking hypothesis is assuming that the brain response is increased for the duration of an event. This could be modeled using a step function instead of impulses. Yet another type of hypothesis might concern a modulation of another predictor. For example, the hypothesis that the magnitude of the neural representation of the speech envelope is modulated by how linguistically informative a given segment is can be implemented by scaling the speech envelope according to each phoneme's linguistic surprisal (Donhauser and Baillet, 2020).

Source localization

While EEG data is often analyzed at senser space, when data is recorded from a dense enough sensor array, neural source localization adds information about the neural origin of different brain responses. In many cases, source localization can also improve the signal-to-noise ratio of a specific response, because it acts as a spatial filter, aggregating information across sensors that is relevant for a given brain region.

A straight-forward extension of the approach described here is to apply a linear inverse solution to the continuous data, and apply mTRF analyses to the virtual current dipoles (Brodbeck et al., 2018b). For this purpose, Eelbrain contains functions that directly convert MNE-Python source estimate objects to `NDVars`. A more advanced approach is the Neuro-Current Response Function technique, which models virtual current dipoles as individual convolution and thus performs deconvolution and source localization jointly. This approach estimates an mTRF model for each virtual dipole that collectively predicts MEG measurements in sensor space (Das et al., 2020). These hybrid approaches can differentiate responses related to processing and comprehension of continuous stimuli such as speech, language etc., anatomically as well as temporally and thus can provide a way to investigate potential hierarchical models of sensory and cognitive processing involving multiple anatomical regions.

Further applications

While cortical processing of speech has been a primary application for deconvolution analysis, the technique has potential applications in any domain where stimuli unfold in time, and has already been successfully applied to music perception (Di Liberto et al., 2020; Leahy et al., 2021) and subcortical auditory processing (Maddox and Lee, 2018). Furthermore, deconvolution analysis as discussed here assumes that TRFs are static across time. However, this is not always a valid assumption. For example, in multi-talker speech, TRFs to speech features change as a function of whether the listener attends to the given speech stream or not (Ding and Simon, 2012; Brodbeck et al., 2018a; Broderick et al., 2018). Thus, moment-to-moment fluctuations in attention might be associated with corresponding changes in the TRFs. While modeling this is a highly complex problem, with an even larger number of degrees of freedom, some initial progress has been made towards estimating deconvolution models with dynamic TRFs that can change over time (Babadi et al., 2010; Miran et al., 2018; Presacco et al., 2019).

References

Alday PM. 2019. M/EEG analysis of naturalistic stories: a review from speech to language processing. *Lang Cogn Neurosci* **34**:457–473. doi:10.1080/23273798.2018.1546882

Babadi B, Kalouptsidis N, Tarokh V. 2010. SPARLS: The Sparse RLS Algorithm. *IEEE Trans Signal Process* **58**:4013–4025. doi:10.1109/TSP.2010.2048103

Bhattasali S, Brennan J, Luh W-M, Franzluebbers B, Hale J. 2020. The Alice Datasets: fMRI & EEG Observations of Natural Language ComprehensionProceedings of the 12th Conference on Language Resources and Evaluation. Presented at the LREC 2020. Marseille. pp. 120–125.

Brennan JR, Hale JT. 2019. Hierarchical structure guides rapid linguistic predictions during naturalistic listening. *PLOS ONE* **14**:e0207741. doi:10.1371/journal.pone.0207741

Brodbeck C, Bhattasali S, Heredia AC, Resnik P, Simon JZ, Lau E. 2021. Parallel processing in speech perception: Local and global representations of linguistic context. *bioRxiv* 2021.07.03.450698. doi:10.1101/2021.07.03.450698

Brodbeck C, Hong LE, Simon JZ. 2018a. Rapid Transformation from Auditory to Linguistic Representations of Continuous Speech. *Curr Biol* **28**:3976-3983.e5. doi:10.1016/j.cub.2018.10.042

Brodbeck C, Jiao A, Hong LE, Simon JZ. 2020. Neural speech restoration at the cocktail party: Auditory cortex recovers masked speech of both attended and ignored speakers. *PLOS Biol* **18**:e3000883. doi:10.1371/journal.pbio.3000883

Brodbeck C, Presacco A, Simon JZ. 2018b. Neural source dynamics of brain responses to continuous stimuli: Speech processing from acoustics to comprehension. *NeuroImage* **172**:162–174. doi:10.1016/j.neuroimage.2018.01.042

Brodbeck C, Simon JZ. 2020. Continuous speech processing. *Curr Opin Physiol* **18**:25–31. doi:10.1016/j.cophys.2020.07.014

Broderick MP, Anderson AJ, Liberto GMD, Crosse MJ, Lalor EC. 2018. Electrophysiological Correlates of Semantic Dissimilarity Reflect the Comprehension of Natural, Narrative Speech. *Curr Biol* **28**:803-809.e3. doi:10.1016/j.cub.2018.01.080

Crosse MJ, Liberto D, M G, Bednar A, Lalor EC. 2016. The Multivariate Temporal Response Function (mTRF) Toolbox: A MATLAB Toolbox for Relating Neural Signals to Continuous Stimuli. *Front Hum Neurosci* **10**. doi:10.3389/fnhum.2016.00604

Das P, Brodbeck C, Simon JZ, Babadi B. 2020. Neuro-current response functions: A unified approach to MEG source analysis under the continuous stimuli paradigm. *NeuroImage* **211**:116528. doi:10.1016/j.neuroimage.2020.116528

Daube C, Ince RAA, Gross J. 2019. Simple Acoustic Features Can Explain Phoneme-Based Predictions of Cortical Responses to Speech. *Curr Biol* **29**:1924-1937.e9. doi:10.1016/j.cub.2019.04.067

David SV, Mesgarani N, Shamma SA. 2007. Estimating sparse spectro-temporal receptive fields with natural stimuli. *Netw Comput Neural Syst* **18**:191–212. doi:10.1080/09548980701609235

David SV, Shamma SA. 2013. Integration over Multiple Timescales in Primary Auditory Cortex. *J Neurosci* **33**:19154–19166. doi:10.1523/JNEUROSCI.2270-13.2013

Di Liberto GM, Pelofi C, Bianco R, Patel P, Mehta AD, Herrero JL, de Cheveigné A, Shamma S, Mesgarani N. 2020. Cortical encoding of melodic expectations in human temporal cortex. *eLife* **9**:e51784. doi:10.7554/eLife.51784

Di Liberto GM, O'Sullivan JA, Lalor EC. 2015. Low-Frequency Cortical Entrainment to Speech Reflects Phoneme-Level Processing. *Curr Biol* **25**:2457–2465. doi:10.1016/j.cub.2015.08.030

Ding N, Chatterjee M, Simon JZ. 2014. Robust cortical entrainment to the speech envelope relies on the spectro-temporal fine structure. *NeuroImage* **88**:41–46. doi:10.1016/j.neuroimage.2013.10.054

Ding N, Simon JZ. 2012. Emergence of neural encoding of auditory objects while listening to competing speakers. *Proc Natl Acad Sci U S A* **109**:11854–9. doi:10.1073/pnas.1205381109

Donhauser PW, Baillet S. 2020. Two Distinct Neural Timescales for Predictive Speech Processing. *Neuron* **105**:385-393.e9. doi:10.1016/j.neuron.2019.10.019

Etard O, Kegler M, Braiman C, Forte AE, Reichenbach T. 2019. Decoding of selective attention to continuous speech from the human auditory brainstem response. *NeuroImage* **200**:1–11. doi:10.1016/j.neuroimage.2019.06.029

Fiedler L, Wöstmann M, Graversen C, Brandmeyer A, Lunner T, Obleser J. 2017. Single-channel in-ear-EEG detects the focus of auditory attention to concurrent tone streams and mixed speech. *J Neural Eng* **14**:036020. doi:10.1088/1741-2552/aa66dd

Fiedler L, Wöstmann M, Herbst SK, Obleser J. 2019. Late cortical tracking of ignored speech facilitates neural selectivity in acoustically challenging conditions. *NeuroImage*

186:33–42. doi:10.1016/j.neuroimage.2018.10.057

Fishbach A, Nelken I, Yeshurun Y. 2001. Auditory Edge Detection: A Neural Model for Physiological and Psychoacoustical Responses to Amplitude Transients. *J Neurophysiol* **85**:2303–2323. doi:10.1152/jn.2001.85.6.2303

Freckleton RP. 2002. On the misuse of residuals in ecology: regression of residuals vs. multiple regression. *J Anim Ecol* **71**:542–545. doi:10.1046/j.1365-2656.2002.00618.x

Friston K, Ashburner J, Kiebel S, Nichols T, Penny W, editors. 2007. Statistical Parametric Mapping, Statistical Parametric Mapping. Elsevier.

Gillis M, Vanthornhout J, Simon JZ, Francart T, Brodbeck C. 2021. Neural markers of speech comprehension: measuring EEG tracking of linguistic speech representations, controlling the speech acoustics. *bioRxiv* 2021.03.24.436758. doi:10.1101/2021.03.24.436758

Gramfort A, Luessi M, Larson E, Engemann DA, Strohmeier D, Brodbeck C, Parkkonen L, Hämäläinen MS. 2014. MNE software for processing MEG and EEG data. *NeuroImage* **86**:446–460. doi:10.1016/j.neuroimage.2013.10.027

Hamilton LS, Huth AG. 2020. The revolution will not be controlled: natural stimuli in speech neuroscience. *Lang Cogn Neurosci* **35**:573–582. doi:10.1080/23273798.2018.1499946

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. 2020. Array programming with NumPy. *Nature* **585**:357–362. doi:10.1038/s41586-020-2649-2

Hastie T, Taylor J, Tibshirani R, Walther G. 2007. Forward stagewise regression and the monotone lasso. *Electron J Stat* **1**:1–29. doi:10.1214/07-EJS004

Heeris J. 2018. Gammatone Filterbank Toolkit.

Kulasingham JP, Brodbeck C, Presacco A, Kuchinsky SE, Anderson S, Simon JZ. 2020. High gamma cortical processing of continuous speech in younger and older listeners. *NeuroImage* **222**:117291. doi:10.1016/j.neuroimage.2020.117291

Lalor EC, Foxe JJ. 2010. Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution. *Eur J Neurosci* **31**:189–193. doi:10.1111/j.1460-9568.2009.07055.x

Lalor EC, Pearlmutter BA, Reilly RB, McDarby G, Foxe JJ. 2006. The VESPA: A method for the rapid estimation of a visual evoked potential. *NeuroImage* **32**:1549–1561. doi:10.1016/j.neuroimage.2006.05.054

Leahy J, Kim S-G, Wan J, Overath T. 2021. An Analytical Framework of Tonal and Rhythmic Hierarchy in Natural Music Using the Multivariate Temporal Response Function. *Front Neurosci* **0**. doi:10.3389/fnins.2021.665767

Maddox RK, Lee AKC. 2018. Auditory Brainstem Responses to Continuous Natural Speech in Human Listeners. *eneuro* **5**:ENEURO.0441-17.2018. doi:10.1523/ENEURO.0441-17.2018

Maris E, Oostenveld R. 2007. Nonparametric statistical testing of EEG- and MEG-data. *J Neurosci Methods* **164**:177–190. doi:10.1016/j.jneumeth.2007.03.024

McAuliffe M, Socolof M, Mihuc S, Wagner M, Sonderegger M. 2017. Montreal Forced Aligner: Trainable Text-Speech Alignment Using KaldiInterspeech 2017. Presented at the Interspeech 2017. ISCA. pp. 498–502. doi:10.21437/Interspeech.2017-1386

Miran S, Akram S, Sheikhattar A, Simon JZ, Zhang T, Babadi B. 2018. Real-Time Tracking of

Selective Auditory Attention From M/EEG: A Bayesian Filtering Approach. *Front Neurosci* **12**. doi:10.3389/fnins.2018.00262

Nichols TE, Holmes AP. 2002. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Hum Brain Mapp* **15**:1–25. doi:10.1002/hbm.1058

Nunez PL, Srinivasan R. 2006. Electric Fields of the Brain: The Neurophysics of EEG, 2nd ed. Oxford: Oxford University Press.

Patterson RD, Robinson K, Holdsworth J, McKeown D, Zhang C, Allerhand M. 1992. Complex Sounds and Auditory ImagesAuditory Physiology and Perception. Elsevier. pp. 429–446. doi:10.1016/B978-0-08-041847-6.50054-X

Presacco A, Miran S, Babadi B, Simon JZ. 2019. Real-Time Tracking of Magnetoencephalographic Neuromarkers during a Dynamic Attention-Switching Task2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 4148–4151. doi:10.1109/EMBC.2019.8857953

R Core Team. 2021. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Reback J, Jbrockmendel, McKinney W, Van Den Bossche J, Augspurger T, Cloud P, Hawkins S, Gfyoung, Sinhrks, Roeschke M, Klein A, Terji Petersen, Tratner J, She C, Ayd W, Hoefler P, Naveh S, Garcia M, Schendel J, Hayden A, Saxton D, Shadrach R, Gorelli ME, Jancauskas V, Fangchen Li, Attack68, McMaster A, Battiston P, Skipper Seabold, Kaiqi Dong. 2021. Pandas. Zenodo. doi:10.5281/ZENODO.3509134

Salverda AP, Dahan D, McQueen JM. 2003. The role of prosodic boundaries in the resolution of lexical embedding in speech comprehension. *Cognition* **90**:51–89. doi:10.1016/S0010-0277(03)00139-2

Smith SM, Nichols TE. 2009. Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. *NeuroImage* **44**:83–98. doi:10.1016/j.neuroimage.2008.03.061

Theunissen FE, David SV, Singh NC, Hsu A, Vinje WE, Gallant JL. 2001. Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Netw Comput Neural Syst* **12**:289–316. doi:10.1080/net.12.3.289.316

Vallat R. 2018. Pingouin: statistics in Python. *J Open Source Softw* **3**:1026. doi:10.21105/joss.01026

Weissbart H, Kandylaki KD, Reichenbach T. 2020. Cortical Tracking of Surprisal during Continuous Speech Comprehension. *J Cogn Neurosci* **32**:155–166. doi:10.1162/jocn_a_01467