Linear Modeling of Neurophysiological Responses to Naturalistic Stimuli: Methodological Considerations for Applied Research

Michael J. **Crosse**[1,2,*], Nathaniel J. **Zuk**[3,*], Giovanni M. **Di Liberto**[4,5], Aaron R. **Nidiffer**[6], Sophie **Molholm**[2], and Edmund C. **Lalor**[6,†]

[1]X, The Moonshot Factory, Mountain View, CA
[2]Department of Pediatrics and Department of Neuroscience, Albert Einstein College of Medicine, Bronx, NY
[3]Edmond & Lily Safra Center for Brain Sciences, Hebrew University, Jerusalem, Israel
[4]Trinity Centre for Biomedical Engineering, Trinity College Institute of Neuroscience, Dept of Mechanical, Manufacturing and Biomedical Engineering, Trinity College, The University of Dublin, Ireland
[5]School of Electrical and Electronic Engineering and UCD Centre for Biomedical Engineering, University College Dublin, Ireland
[6]Department of Biomedical Engineering and Department of Neuroscience, University of Rochester, Rochester, NY

Abstract

Cognitive neuroscience has seen an increase in the use of linear modelling techniques for studying the processing of natural, environmental stimuli. The availability of such computational tools has prompted similar investigations in many clinical domains, facilitating the study of cognitive and sensory deficits within an ecologically relevant context. However, studying clinical (and often highly-heterogeneous) cohorts introduces an added layer of complexity to such modelling procedures, leading to an increased risk of improper usage of such techniques and, as a result, inconsistent conclusions. Here, we outline some key methodological considerations for applied research and include worked examples of both simulated and empirical electrophysiological (EEG) data. In particular, we focus on experimental design, data preprocessing and stimulus feature extraction, model design, training and evaluation, and interpretation of model weights. Throughout the paper, we demonstrate how to implement each stage in MATLAB using the *mTRF-Toolbox* and discuss how to address issues that could arise in applied cognitive neuroscience research. In doing so, we highlight the importance of understanding these more technical points for experimental design and data analysis, and provide a resource for applied and clinical researchers investigating sensory and cognitive processing using ecologically-rich stimuli.

Keywords: temporal response function, TRF, neural encoding, neural decoding, clinical and translational neurophysiology, electrophysiology, EEG.

[*] These authors contributed equally to this work.
[†] e-mail: edmund_lalor@urmc.rochester.edu (E.C.L.)

Introduction

A core focus of cognitive neuroscience is to identify neural correlates of human behavior, with the intention of understanding cognitive and sensory processing. Such correlates can be used to explicitly model the functional relationship between some “real world” parameters describing a stimulus or person’s behavior and the related brain activity. In particular, linear modelling techniques have become ubiquitous within cognitive neuroscience because they provide a means of studying the processing of dynamic sensory inputs such as natural scenes and sounds (Wu et al., 2006; Holdgraf et al., 2017). Unlike event-related potentials (ERPs) – which are a direct measurement of the average neural response to a discrete event – linear models seek to capture how changes in a stimulus dimension or cognitive state are linearly reflected in the recorded brain activity. In other words, we model the outputs as a linear combination (i.e., weighted sum) of the inputs. This enables researchers to conduct experiments using ecologically relevant stimuli that are more engaging and more representative of real-world scenarios. This contrasts with current standard practices in which discrete stimuli are presented repeatedly in a highly artificial manner. Moreover, the simplicity of linear models enables researchers to interpret the model weights neurophysiologically, providing insight into the neural encoding process of naturalistic stimuli (Haufe et al., 2014; Kriegeskorte and Douglas, 2019).

The uptake in linear modelling techniques in cognitive neuroscience has led to a similar adoption in the applied and translational neurosciences. This has greatly facilitated the study of naturalistic sensory processing in various clinical cohorts such as individuals with autism spectrum disorder (Frey et al., 2013) and dyslexia (Power et al., 2013; Di Liberto et al., 2018b). However, studying clinical cohorts raises important issues when constructing and interpreting linear models. For example, particular care is required when performing group comparisons of the model weights and evaluating model performance. Furthermore, linear modeling poses challenges and considerations that are not typical for other types of electrophysiology analysis. As a model, it is meant first and foremost to quantify the functional relationship between the stimulus features of interest and the recorded neural response. Modeling electrophysiological data is non-trivial because neighboring time samples and channels are not independent of each other, so standard methods for quantifying the significance of the fit cannot be used. Furthermore, the interpretation of the results must take into careful consideration the particular preprocessing steps applied, which can have major effects on the response patterns obtained with linear modeling, especially with respect to filtering, normalization and stimulus representation (Holdgraf et al., 2017; de Cheveigné and Nelken, 2019). Here, we wish to provide guidance and intuition on such procedures and, in particular, to promote best practices in applying these methods in clinical studies.

In this review, we will step through the stages involved in designing and implementing neuroscientific experiments with linear modeling in mind. First, we discuss experimental design considerations for optimizing model performance. Second, we discuss data preprocessing and stimulus feature extraction techniques relevant to linear modeling. Third, we discuss model design choices and their use cases. Fourth, we review how to appropriately train and test models as well as evaluate the significance of model performance. Fifth, we discuss considerations for comparing models generated using multiple stimulus representations. Sixth, we discuss the neurophysiological interpretation of linear model weights. Finally, we discuss what can go wrong when using linear models for applied neurophysiology research.

In each section, via an example experiment, we will also introduce issues that are relevant to clinical research. Because linear modeling is commonly used to study the neural processing of natural speech (for reviews, see Ding and Simon, 2014; Holdgraf et al., 2017; Obleser and Kayser, 2019), these examples are based on a speech study previously conducted by some of the authors, but the methods we describe generalize to many other clinical groups, paradigms, and stimulus types. The researcher should modify the experimental design, preprocessing and model design steps according to their own research questions. Likewise, our focus will be on the linear modeling of EEG data, but these methods can be applied to other neurophysiological data types, such as MEG, ECoG and fMRI. When discussing model implementation, we will make specific reference to the mTRF-Toolbox, which can be found on github (https://github.com/mickcrosse/mTRF-Toolbox). All functions referenced in this article were from version 3.0. While we do not elaborate on the technical details of the mTRF-Toolbox (for that we point the reader to Crosse et al. (2016a)), we do provide example code and briefly walk the reader through its implementation.

Example Experiment

The example experiment we will describe is based on a previous study performed by some of the co-authors in this review (Di Liberto et al., 2018b). Individuals with dyslexia (our clinical group) display a specific behavioral deficit in the processing of speech sounds (i.e., a phonological deficit), while having intact general acoustic processing (Vellutino et al., 2004; Di Liberto et al., 2018b). We hypothesize that observed phonological deficits can be explained by weaker phonetic encoding.

To test our hypothesis, we plan to measure how well phonetic features are represented in the ongoing brain activity of participants with dyslexia compared to a control group. More specifically, we will quantify how much a model that represents phonetic features improves the ability to predict EEG data over a model based on acoustic features alone (i.e., the spectrogram). We hypothesize that the predictive contribution from the phonetic model is reduced in participants with dyslexia, reflective of impaired neural tracking of phonetic features, while the contributions of acoustics are comparable between groups. To be clear, while it is inspired by a real study, the example experiment we discuss in this paper is merely a toy experiment for didactic purposes.

Experimental design

One of the benefits of employing linear models for EEG analysis is the ability to use dynamic and naturalistic stimuli. Additionally, it allows the experimenter to study sensory processing in an ecologically-relevant context, and it also provides researchers the opportunity to design experiments that are more engaging for the participants. This can potentially improve the quality of the data collected as well as the reliability of the researcher's findings. Certain factors should be considered when designing naturalistic experiments.

Use subject-relevant stimulus material. This is primarily relevant to speech studies and is important for ensuring subject compliance with the task, particularly when studying younger cohorts and individuals with neurological disorders or developmental disabilities. For example, it is important when choosing an audiobook or movie, that it is 1) age-relevant (e.g., a children's story versus an adult's podcast), 2) content-relevant (a quantum physics lecture may not be everyone's cup of tea), and 3) language-relevant (speaker dialect and even accent may impact early-stage processing across participants/groups differentially). It may in some situations be necessary to create such content from scratch by recording a native speaker reading the chosen material aloud. However, there are also publicly available stimulus databases such as MUSAN: an annotated corpus of continuous speech, music and noise (Snyder et al., 2015), and TCD-TIMIT: a phonetically rich corpus of continuous audiovisual speech (Harte and Gillen, 2015).

Use a well-balanced stimulus set. It is important to consider the frequency of occurrence of particular stimulus features that are relevant to the study (e.g., spectral or phonetic features). For example, choosing stimulus material that contains only a few instances of particular phonemes will make it difficult to reliably model the neural response to such phonemes without overfitting to the noise on those examples. This can be avoided by employing phonetically balanced stimuli, such as the aforementioned TCD-TIMIT corpus

(Harte and Gillen, 2015), or in a post hoc manner by focusing the analysis on a subset of the data, i.e., only the features that are equally represented or only the time segments where the stimuli are well balanced. It is also best to work with longer stimuli that are preferably broadband or quasi-periodic (e.g., speech or music recordings). Linear modeling can produce ambiguous results if the stimulus is perfectly periodic since periodicity can result in artificially periodic-looking evoked responses which can also increase difficulties with quantifying the accuracy of the model.

In addition, to enhance the model's ability to disambiguate these response types and better generalize to novel stimuli, one might consider how to incorporate additional acoustic variability in one's stimuli, independent of the linguistic content. This could be accomplished by including multiple speakers with substantially different spectral profiles (e.g., both male and female speakers), as well as speakers who provide a more dynamic range in prosody and intonation across the speech content (e.g., trained actors or media presenters). Models that are trained on a broader range of stimuli are less likely to overfit to stimulus features that are not of interest to the researcher (such as speaker identity, sex, or location), but may perform slightly worse on average. Such decisions should be based on the researcher's overall goals.

When considering your stimuli, we also suggest adopting an open mind with respect to possible future analyses. Choosing materials that are rich in other features that can be modeled (e.g., semantic content, prosody, temporal statistics) can provide fruitful opportunities for re-using your data to tackle new questions beyond those planned in your current study (fans of Dr. Seuss and James Joyce beware!).

Collect enough training data. In order to train a model that generalizes well to new data, it is crucial to consider how much training data is required, or in other words, how much new stimulus material it is necessary to have. For most purposes, we recommend collecting a minimum of 10 to 20 minutes of data per condition, although more data may be required for larger, multivariate models (e.g., spectrogram models) or when features are sparsely represented (e.g., the onsets of content words). While it is feasible to construct high-quality models from many short (<5 s) stimulus sequences, such as individual words or sentences, it is preferable to use longer (>30 s) stimulus passages because it reduces the number of large stimulus onset responses in the neural data, which tend to obscure feature-specific responses of interest (see *EEG preprocessing* for tips on avoiding this).

While more data is always desirable for model training, longer recording sessions can cause subject fatigue, compromising their ability to concentrate, particularly in children, older adults, or clinical cohorts. Reduced attentional states can negatively impact the neural tracking of stimuli and as a result model performance (Ding and Simon, 2012; O'Sullivan et al., 2015). Recording time can be shortened without compromising model quality by a) removing silent periods in the speech stimuli greater than a certain duration in order to maximize information rate (e.g., >300 ms; Ding and Simon, 2013) or b) using subject-independent models, i.e., models constructed from the data of multiple subjects (see *Model design*; Di Liberto and Lalor, 2017). This issue applies when considering how many different experimental conditions to include. When a large number of conditions is unavoidable we recommend splitting the recordings into multiple sessions, as inter-session model performance is typically very reliable (Mahajan et al., in prep). Overall, the researcher should carefully balance the tradeoff between data quality and quantity in a way that meets their needs.

Use active task designs. Although naturalistic stimuli tend to be inherently more engaging than artificial stimuli, the use of prolonged natural stimulus sequences can still induce fatigue which can negatively impact data quality and results. To ensure continuous engagement with the stimulus content, we recommend including an appropriate behavioral task. This can consist of answering comprehension questionnaires immediately after the end of each trial or detecting intermittent targets or anomalies in the stimulus (e.g., respond to a certain word, nonsense word or acoustic perturbation)[†]. The other advantage of including a task is that, in addition to potentially improving the quality of the data, the researcher will have valuable behavioral data to go with it, such as measures of comprehension, intelligibility, detection accuracy or response time. Even tasks that are inherently active, such as auditory attention experiments, can be greatly enhanced by the addition of appropriate psychophysical tasks.

Example experiment

With these considerations in mind, we chose for our example experiment two professionally-recorded audiobooks, each featuring a different Sherlock Holmes adventure. One story is read by a male actor, the other by a female actor, giving us increased acoustic variation across phoneme repetitions. Both stories last approximately one hour, giving us two hours of EEG data per subject with which to fit our models. The stimuli are presented to our subjects in 1-minute-long continuous passages, striking a nice balance between reducing onset effects while still providing regular breaks to minimize fatigue and discomfort. During stimulus presentation, we record 128-channel EEG data (sampled at 512 Hz) for each of our trials

[†] It is important to use an appropriate task to avoid the subject engaging with the wrong aspect of the stimulus. For example, if we wish them to engage with the language content, then using a speech target, such as a nonsense word, is better than them listening out for a physical acoustic anomaly.

(60 × 1-minute trials × 2 stories; Figure 1A). We collect these data from 30 individuals with dyslexia and 30 sex, age, and IQ-matched control subjects.

EEG preprocessing

Prior to constructing our model, it may be necessary to clean and preprocess the EEG data. In our experience, linear modeling is fairly robust to sparse artifacts, such as eyeblinks, which are usually uncorrelated with the stimulus dynamics/features of interest. However, particularly noisy data can be difficult to work with, and neural recordings such as EEG are notoriously noisy. On the other hand, aggressive filtering of the data is unfortunately common and can exacerbate existing artifacts, producing spurious oscillatory modulations or ringing in the EEG (for an in-depth review, see de Cheveigné and Nelken, 2019). In this section, we generally recommend a more conservative EEG preprocessing strategy to avoid this, but also make suggestions for a more liberal approach as needed for noisier datasets and less cooperative cohorts (e.g., infants or certain clinical groups). Ideally, a single set of preprocessing parameters – defined by the noisiest cohort/data – should be applied to all participants within a given study.

In general, we recommend the following preprocessing steps in order:

1. High-pass filter (HPF) the EEG data to remove any unwanted DC shifts or slow drift potentials (<1Hz) that may be present, for example, due to amplifier DC components or electrode junction potentials caused by sweating. Note, higher-order filters with sharp roll-offs can introduce significant artifacts in the time domain and thus should be avoided (de Cheveigné and Nelken, 2019). Additionally, we recommend zero-phase-shift filters to prevent temporal shifts that can arise during filtering, which would affect the temporal relationship between the stimulus and response when modeling. However, zero-phase filters are non-causal, which can affect the interpretation of model weights prior to zero lag (see *Interpretation of model weights*, and de Cheveigné and Nelken, 2019). To remove unwanted low-frequency artifacts, we recommend the following HPF parameters for cooperative subjects: cutoff frequency in the range [0.1, 0.5] Hz, order ≤ 3. For less cooperative subjects, we recommend the following: cutoff frequency in the range [0.5, 1.0] Hz, order ≤ 5. Note, for modeling information processing at specific EEG frequency bands (e.g., alpha, beta, low gamma), one may wish to use much higher HPF cutoffs as part of a band-pass filter.

2. Low-pass filter (LPF) the EEG data to remove any unwanted high frequency noise that may be present, for example, due to muscle contractions or environmental interference such as 50/60-Hz line noise. Linear modeling tends to automatically filter the data in such a way that focuses on the most relevant energy range (usually lower-frequencies) for stimulus-based prediction (de Cheveigné et al., 2018), however, a LPF step can help in particular with nosier datasets. Similar to HPF, we also recommend using zero-phase filters. To remove unwanted high-frequency artifacts, we suggest the following LPF parameters for cooperative subjects: cutoff frequency in the range [20, 40] Hz, order ≤ 3. For less cooperative subjects, we recommend the following: cutoff frequency in the range [10, 30] Hz, order ≤ 5. Again, for modeling information processing at specific EEG frequency bands (e.g., delta, theta, alpha), one may wish to use much lower LPF cutoffs as part of a band-pass filter.
3. Remove or interpolate EEG channels with relatively high variance, significant noise/artifacts, or that have been compromised due to bridging. We recommend limiting channel interpolation to < 10% of all channels, any more than that may warrant discarding the entire segment of data altogether. As a sanity check, if the newly interpolated data has a high correlation with the original, discarded data (i.e., Pearson's $r > 0.5$), then the original data may in fact be of good quality and it may not be necessary to interpolate them.
4. Re-reference the EEG data to an appropriate channel(s) to enhance neural activity in a region of interest or recover the common mode rejection ratio (CMRR; necessary for EEG systems such as BioSemi). While model predictions may be unaffected by re-referencing, it can enhance the interpretation of the model weights when done correctly. Mastoid references tend to emphasize responses to auditory stimuli over fronto-central scalp while more frontal references tend to emphasize responses to visual stimuli over occipital scalp. Alternatively, an average reference (i.e., the mean of all channels) is a good choice if one does not want to emphasize the activity of a particular region over another, as might be the case in multisensory experiments (for review, see Murray et al., 2008). However, when using an average reference, ensure that noisy channels are first removed or interpolated (see step 3).
5. Downsample the data to make model fitting and testing more computationally efficient. This is particularly important for larger multivariate datasets which can have long training times. Prior to downsampling, low-pass filter the data well below the Nyquist frequency (i.e., below half the desired sampling rate) to prevent aliasing. Note that some downsampling functions have an anti-aliasing filter built in (e.g., `resample()` in MATLAB). If a LPF is applied in step 2, it can function

as an anti-aliasing filter provided it is below the Nyquist frequency; an additional antialiasing filter constitutes an unnecessary filtering operation that introduces additional filtering artifacts. After downsampling, the sample rate should be at least 2 times the highest frequency of interest. For quantitative modelling analyses, it is possible to downsample to as low as 50 Hz, as most of the EEG frequency content is below 25 Hz. For more qualitative analyses such as interpretation of model weights or for visualization purposes, a higher resolution of >100 Hz may be desired.

6. [OPTIONAL] A sparsity-driven artifact rejection method, such as independent components analysis (ICA; Hyvärinen and Oja, 2000), can be used to remove distinct artifacts such as eye-blinks, facial and neck movements, and line noise.
7. Remove the first 500 to 1000 ms of EEG data at the start of every trial to avoid modelling the response to the stimulus onset. Such responses are typically much larger than those that track the ongoing changes in the speech signal. For shorter (<5 s) stimuli, this may not be feasible but can be helped by avoiding using stimuli with abrupt onsets or by having an initial ramp up in the stimulus intensity.
8. [OPTIONAL] Normalizing the EEG data is not always necessary but there are certain circumstances where this may be desirable. For example, it can also be useful to normalize EEG data when comparing the magnitude of model weights across datasets collected at different testing locations or using different equipment to remove any inherent differences in amplifier gain. Note, this must be done with extreme care, especially when comparing different clinical groups. A common normalization method is to standardize (or z-score) the EEG by subtracting its mean and dividing by its standard deviation. It is important to ensure that if there are multiple EEG files within a recording session, normalization is done with respect to the global mean and standard deviation to ensure consistency across trials (however, one may wish to omit the test data to avoid overfitting). Additionally, this should not be done electrode-by-electrode in order to maintain the relative magnitudes between channels. Normalization of EEG data can also be used to obtain meaningful units of measure when computing model weights, for example, by scaling the data by the EEG amplifier microvolts/bits conversion ratio to obtain microvolt units (see Lalor et al., 2006). This step must be done in tandem with the corresponding stimulus normalization step described in *Stimulus representation* and requires careful quantification of certain physical stimulus parameters. In general, we recommend avoiding normalizing the EEG data unless it is deemed completely necessary.

Example experiment

Since in our example we are interested in how EEG encodes the speech envelope and phonetic features, we base our preprocessing strategy on previous speech research. Neural encoding of natural speech occurs primarily in the delta (0.5–4 Hz) and theta (4–8 Hz) EEG frequency bands (Poeppel, 2003; Giraud and Poeppel, 2012; Ding and Simon, 2014), so we elect to filter our EEG data between 0.2 and 40 Hz using a 3rd order Butterworth filter. Note, such conservative filtering is considered a preprocessing step and leaves open the possibility of filtering the data further to explore specific hypotheses about how activity in different frequency bands (e.g., delta, theta, alpha, beta, low gamma) might differentially reflect the processing of certain features of speech (Ding et al., 2014; Di Liberto et al., 2015). Finally, given that we have filtered our data below 40 Hz, we downsample our data from 512 Hz to 128 Hz in order to optimize efficiency.

Further considerations

These preprocessing steps should be conducted on the entire dataset prior to segmenting it into individual trials (i.e., before extracting the minute-long epochs time-locked to the stimulus triggers). Filtering after data segmentation can introduce edge artifacts that could masquerade as large onset/offset responses and impact subsequent analyses. However, if the EEG data consists of discontinuous trials (i.e., in separate files or pauses in the recording between trials), then filtering should be done separately on each trial because discontinuities at trial boundaries can introduce filter artifacts. However, artifact rejection such as ICA should be done collectively by concatenating all files belonging to the same recording session.

When using a conservative HPF cutoff (e.g., ≤0.1 Hz), you may wish to first conduct ICA on EEG data that was filtered using a higher cutoff (e.g., >1 Hz) and then apply the corresponding ICA weights to the 0.1-Hz HPF data. The reason for this being that ICA works best on data filtered above 1 Hz because it can become biased towards lower frequencies which tend to have greater power and can be contaminated by electrode junction potentials (Winkler et al., 2015). Alternatively, linear denoising source separation (DSS) methods based on the reproducibility of the neural response to a repeated stimulus can be used to enhance the underlying stimulus-driven EEG signal (de Cheveigné and Parra, 2014). However, many linear modeling studies specifically avoid using repeated stimuli in an effort to maximize ecological validity and minimize the effects of prediction. In those cases, linear denoising approaches can be used by attempting to maximize responses that are common across subjects (de Cheveigné et al., 2019).

Stimulus representation

Linear modeling requires the researcher to quantitatively define specific stimulus features that are to be related to the corresponding brain activity. Decisions to quantify the stimulus features a certain way come with their own underlying assumptions. Ultimately, these choices define our hypothesis about how the stimulus is represented in the neural activity, and it will determine the stage of processing along the sensory or cognitive pathway represented by the resulting model.

Choosing stimulus features. Stimulus features are derived from some physical or perceptual parameters that describe the stimulus. Examples of physical parameters range in complexity from intensity of acoustic vibrations and pixel luminance, to motion and spectral content (Figure 1A). Perceptual parameters are more abstract and often relate to how such physical parameters are perceptually mapped to categorical attributes such as words or objects. A linear model is defined by its choice of stimulus features as they determine the level of processing along the sensory or cognitive pathway to be modeled. Such low-level and high-level stimulus features can also be combined to create richer stimulus representations that span multiple processing stages and explain more of the variance in the neural activity (e.g., Di Liberto et al., 2015; O'Sullivan et al., 2017). However, when making decisions about stimulus representations, the resulting number of model parameters relative to the amount of available experimental data should be kept in mind; too many parameters with too few observations can result in overfitting, i.e., the model weights will be optimized to the training data but will generalize poorly to new data (such as left-out trials or subjects) and thus will not be a reliable representation of the neural system under study. This can be avoided by both reducing the number of model parameters and employing a technique known as regularization (see *Model training and testing*).

Non-linear transformations. Another important factor to consider when extracting the stimulus features is that it provides an opportunity to apply biologically relevant non-linear transformations that would otherwise be overlooked by the computations of a linear model. For example, many aspects of human auditory perception, such as loudness and pitch, bear a logarithmic relationship to the corresponding physical properties of sound (Stevens, 1955). Such non-linear relationships between the raw audio signal and the related brain activity can be explicitly incorporated into a linear model by applying an appropriate logarithmic scaling to the relevant stimulus features. This yields a more biologically plausible model that still benefits from the efficiency and interpretability of linear computations. We will step through these considerations in our example experiment, but such parameterizations will be up to the researcher who should consider: 1) their expectations based on prior literature, and 2) an appropriate, and not overparameterized, representation of the features of interest. Considerations here will affect the interpretation of the model weights (for further discussion, see *Interpretation of model weights*).

Normalization. Normalization of stimulus features is a common preprocessing procedure in modeling. It serves to ensure that differences in the scale of features does not influence the magnitude of the feature weights, and results in more consistent parameter tuning across datasets during training. A common normalization method is to z-score the stimulus features by subtracting by its mean and dividing by its standard deviation. As mentioned above, if there are multiple stimulus files, it is important that such normalization is done with respect to the global mean and standard deviation to ensure consistency across trials. Normalization of stimulus features can also be used to obtain meaningful units of measure when computing model weights, for example, by scaling the intensity feature by the frame rate of the original stimulus to obtain microvolt units (see Lalor et al., 2006). This step can be done in tandem with the corresponding EEG normalization step described in *EEG preprocessing*.

Example experiment

In this experiment, we are interested in determining if a phonetic representation of speech explains variance in our EEG beyond that explained by an acoustic representation. The first step is to isolate the particular feature of interest from the stimulus: the “acoustics”, which will be represented by the speech envelope and spectrogram, and the “phonetics”, which will be represented by the timing of phonetic features. In general, it is best to consider prior literature on this topic and describe the method of extracting the features as much as possible when reporting the results. In our example, we will use feature extraction methods based on prior speech literature. To retrieve the spectrogram, the stimulus will be filtered into 32 logarithmically-spaced frequency bands using a gammachirp filterbank to model human auditory frequency perception (Irino and Patterson, 2006). The narrowband envelopes are then computed by taking the moving root-mean square (RMS) over windows of 250 samples to downsample the audio from 16 kHz to 128 Hz to match the rate of the EEG (Lalor and Foxe, 2010). Compression was then applied to the RMS intensity using a logarithmic scaling ($x^{0.3}$) to model human auditory intensity perception (Stevens, 1955). As mentioned above, this non-linear transformation will improve the efficacy of our linear model without increasing its complexity. The broadband envelope is then obtained by summing over frequency bands. This procedure can be implemented in mTRF-Toolbox using the `mTRFenvelope` function. To retrieve the timings of the phonetic features, we use a forced aligner that, given text, will align the onset and offset of each of the phonemes based on the spectrogram (Prosody-aligner, Montreal Forced Aligner) and then map the phoneme timings to their corresponding phonetic features.

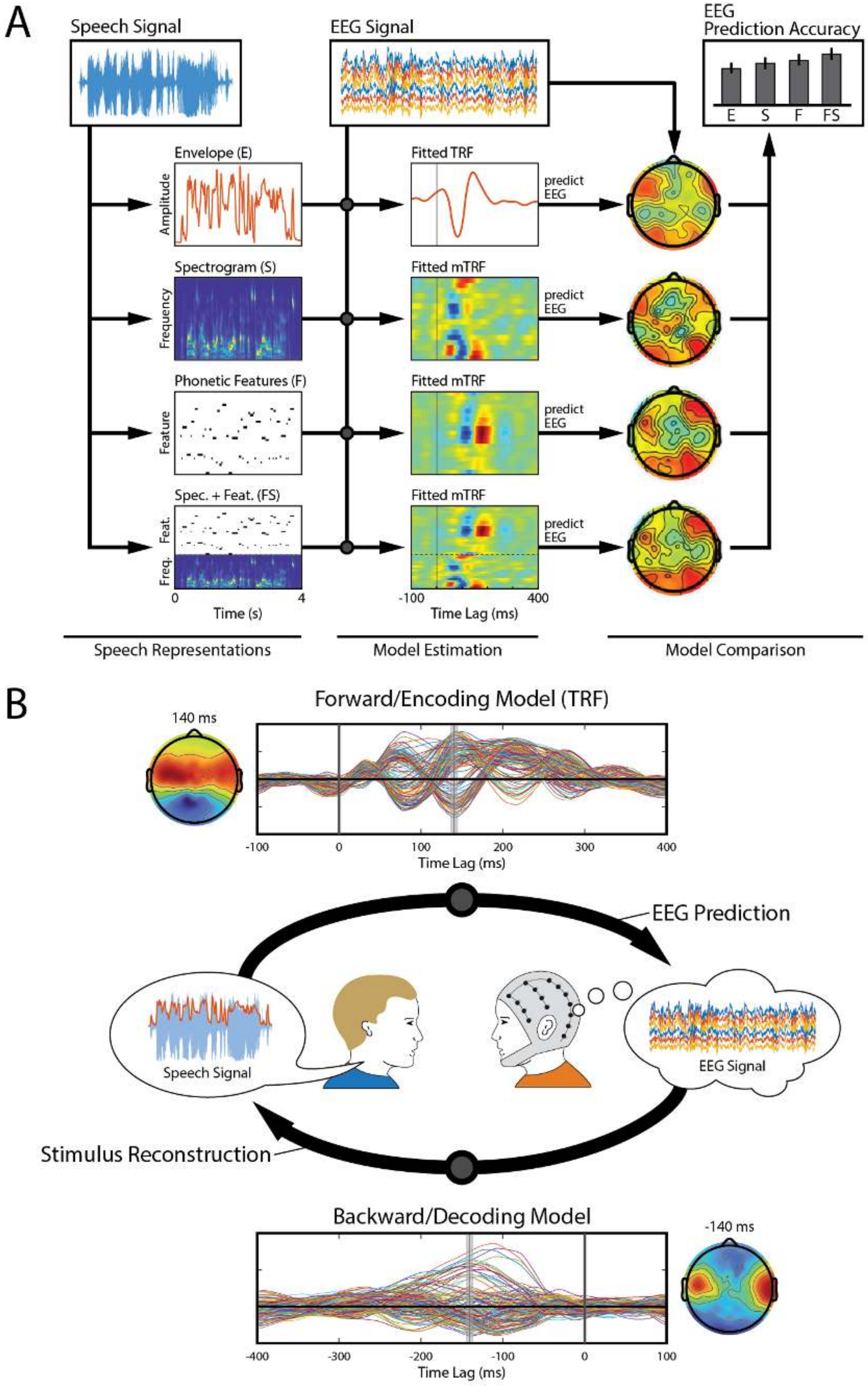
A
Speech Signal
EEG Signal
EEG
Prediction Accuracy
E
S
F
FS
Envelope (E)
Amplitude
Fitted TRF
predict
EEG
Spectrogram (S)
Frequency
Fitted mTRF
predict
EEG
Phonetic Features (F)
Feature
Fitted mTRF
predict
EEG
Spec. + Feat. (FS)
Feat.
Freq.
Fitted mTRF
predict
EEG
0
4
Time (s)
-100
400
Time Lag (ms)
Speech Representations
Model Estimation
Model Comparison
B
Forward/Encoding Model (TRF)
140 ms
-100
0
100
200
300
400
Time Lag (ms)
EEG Prediction
Speech Signal
EEG Signal
Stimulus Reconstruction
Backward/Decoding Model
-140 ms
-400
-300
-200
-100
0
100
Time Lag (ms)

FIGURE 1 | Linear modeling framework and comparison across a family of models. (A) A speech signal can be represented by several different features, such as the envelope, spectrogram, and phonetic features. Each of these features (or combinations of them) can be regressed against brain activity to estimate single- or combined-feature encoding models (TRFs). The TRFs can be used to predict held-out EEG data, channel-by-channel, and the accuracy of those predictions can be measured to assess the quality of the model. Here, we compare model performance across a set of six bilateral auditory-responsive electrodes (adapted from Di Liberto et al., 2015; Crosse et al., 2016a). **(B)** In a series of experiment trials, an observer (right) is presented a stimulus – here, a speaker (left) produces a speech signal (blue time-series shown in the speech bubble), while EEG is recorded simultaneously from their scalp (multi-colored time-series shown in the thought cloud). We can extract any of several features from that stimulus, such as the envelope (red trace). Forward modeling (top arrow) fits a set of weights in an attempt to predict EEG data from a set of stimulus features. Those weights, known as a Temporal Response Function (TRF), are biologically interpretable, akin to a conventional Event Related Potential (ERP). Conversely, backward modeling (bottom arrow) fits a set of weights that map in the reverse direction, known as a decoder, in order to reconstruct a set of stimulus features from the EEG data. While these coefficients are informative, they are not neurophysiologically interpretable in the same way as a TRF.

Model design

In addition to choosing the relevant input/output features for our model, there are a number of design considerations that the researcher must make regarding the model itself. Because we focus on linear models here, we do not need to consider the model architecture. Our first design choice is what direction we wish the model to map between our stimulus features and brain data, i.e., should it be a forward model or a backward model (see Figure 1B)? The second decision is how much temporal context the model should have; should it only map between single time points or integrate over several hundred milliseconds of data? Third, we need to decide whether we want to construct our model using data from one or more subjects in order to optimize generalization of the model. In the following sections, we will discuss the details of each design choice and their use cases. For implementation in mTRF-Toolbox, see Box 1.

Forward models. To paraphrase others (e.g., Carandini et al., 2005), the ultimate test of our understanding of sensory processing (and cognitive processing of sensory stimuli) is our ability to predict neural responses to those stimuli. To test hypotheses about those computations and resulting representations, one can build models that attempt to capture how different features of a sensory stimulus affect one’s ability to predict the neural responses (including at different time lags and on different neural recording channels). These models are known as forward or encoding models,

[3] and in the EEG literature they are commonly referred to as temporal response functions or TRFs (Ding and Simon, 2012). The weights of a TRF describe how the EEG signal on a given recording channel modulates in response to a unit change in a particular stimulus feature. The temporal dynamics of a TRF typically exhibit a close correspondence to those of an ERP (Lalor et al., 2006; Lalor et al., 2009). However, the components of an ERP describe how the EEG signal on a given recording channel modulates in response to the entire stimulus event (that said, ERP researchers often attempt to get at the same information by calculating the difference in ERPs to isolated events that differ only in a specific parameter). Moreover, we can use a TRF to predict the neural response to a given stimulus sequence, and control its smoothness in a way that allows it to generalize well to new data (Haufe et al., 2014; Kriegeskorte and Douglas, 2019). We later discuss ways to ensure confidence in an encoding model's ability to represent the underlying neural response (see *Interpreting the model weights*). Importantly, the interpretability of forward models means that they can be used to both identify processing deficits in particular clinical groups, as well as understand the underlying neurophysiology.

Backward models. An alternative to building forward encoding models is to quantify how well certain stimulus features can be decoded from the neural activity under different experimental conditions or in different subject groups. To do this one can construct a model that maps backwards from brain to stimulus (Mesgarani et al., 2009; Ding and Simon, 2012). Such backward models can be used to reconstruct or decode the stimulus features from the neural activity; thus, they are commonly referred to as decoding models or decoders. However, because these models typically involve optimizing the weighted sum of all neural recording channels simultaneously, the resulting decoder weights are difficult to interpret neurophysiologically (Haufe et al., 2014; Holdgraf et al., 2017). Despite their shortcomings in interpretability, decoders have distinct advantages over encoding models: 1) they do not require pre-selection of neural channels because each channel is weighted according to how informative it is, 2) decoders can utilize independent information on different neural channels to optimally infer stimulus features because they operate on the multichannel neural activity, 3) reconstruction accuracy is usually higher because decoders project to the stimulus domain where we often have direct access to the ground truth (which is not the case for noise-ridden EEG), and 4) decoders can utilize any useful neural information that correlates with the stimulus feature, even if that neural activity did not explicitly encode that feature (Mesgarani et al., 2009). We recommend linear modeling only for reconstructing continuous

[3] In neural systems, the natural flow of information propagates forward from stimulus to brain. Because such models describe how information is encoded in the neural activity, they are also known as encoding models.

stimulus features (e.g., speech envelope or spectrogram), and not for reconstructing discrete stimulus representations (e.g., phoneme time stamps) which require additional nonlinear transformations (e.g., Huth et al., 2016; Zuk et al., 2019). Forward and backward models thus have distinct functions and can be employed in a complementary manner to investigate both qualitative and quantitative research questions.

Time lags. An additional design consideration is the range of time delays or lags to include in our model. We know from the ERP literature that it takes several hundred milliseconds for sensory information to propagate throughout the cortex, with different time lags reflecting different stages of sensory/cognitive processing. Including such time lags in our model allows us to capture the dynamics of the temporal relationship between the stimulus and the neural response, such that the model can utilize relevant information at specific time delays to make better predictions. That means including past stimulus information for making predictions about present neural activity (forward models), and future neural activity for making predictions about present stimulus information (backward models). We recommend using a range of time lags based on the delays at which one expects to see a stimulus-response relationship. This can be determined by referencing previous TRF literature (or even ERP literature), or empirically by testing model performance using different lags. For cortical responses, this is typically on the order of hundreds of milliseconds (e.g., 0 to 300 ms), whereas for auditory brainstem responses this is typically on the order of tens of milliseconds (e.g., 0 to 10 ms). For more specific hypotheses, one may wish to limit the time delays to early (0 to 150 ms) versus late (150 to 300 ms), or look at performance as a function of time using a moving window or single lags (see O'Sullivan et al., 2015; Crosse et al., 2016b). For visualization purposes, it is often desirable to include pre- and post-response lags (e.g., −100 to 400 ms) to show the baseline activity or quantify the noise floor. While using a broader range of lags can often yield better predictions, it also increases the number of parameters which increases susceptibility to overfitting. However, there are techniques to combat overfitting such as regularization (see *Model training and testing*).

Subject dependency. The last consideration when designing a linear model is whether it should be constructed using individual subject data (subject-dependent model) or data from multiple subjects (subject-independent model). Subject-dependent models are more common in the literature because their performance is typically better due to inter-subject variability in the neural responses. However, when it is not possible to collect enough data per subject or per condition, subject-independent models can provide an alternative way of improving model robustness (Di Liberto and Lalor, 2017; Di Liberto et al., 2018b). Note, this approach assumes a certain level of homogeneity within each subject group[4]. There are numerous ways in which to implement subject-independent models: 1) the model is trained on *n*−1 subjects and tested on the data of the held-out subject, 2) the model is trained on all *n* subjects and tested on individual held out trials, 3) a pre-trained subject-independent model is combined with subject-dependent training data to improve model performance (i.e., transfer learning). For more information on how to appropriately partition data for training and testing models, see *Model training and testing*.

Example experiment

For our example experiment, we decide to go with a forward model design for the following reasons: 1) we wish to compare EEG tracking of the acoustics and phonetics in speech, hence evaluating them in the EEG domain makes for a fairer comparison, 2) we wish to understand any potential group differences in speech processing and thus want the model weights to be neurophysiologically interpretable. Based on prior work, we expect to see neural responses to the spectrogram and phonemes within 300 ms (Di Liberto et al., 2015; Brodbeck et al., 2018). As an initial inquiry, we decide to construct our model with time lags between –100 and 400 ms in order to observe the entire TRF timecourse and the pre- and post-TRF baseline. For model evaluation, we restrict the time lags to between 0 and 300 ms (based on empirical testing) to optimize its ability to predict the neural response. Note, we do not merely truncate the original model, but rather retrain the model using this restricted lag range.

Further considerations

While utilizing as many relevant time lags as possible often leads to better model predictions, it can sometimes obscure differences in neural processing between conditions/groups because such differences may occur at specific timepoints along the sensory pathway. For example, if our example clinical group differ only in phonetic processing, then it is possible that earlier acoustic processing remains unimpaired, and models based on a broad range of time lags will yield only a small group effect or even none at all. Using instead a 2 × 2 analysis design, we can examine model performance based on early versus late lags within each of the groups as well as any potential interactions that may exist. Alternatively, a single lag analysis could be used to examine model performance as a function of time lag for each of the groups.

[4] Homogeneity can often be lower in certain clinical populations which may negatively impact the performance of subject-independent models relative to that of the control group. As such this should be taken into consideration when designing the model. On the other hand, subject-independent models can potentially be used to empirically demonstrate such differences in homogeneity.

Box 1. Model design implementation.

Suppose we have a matrix of EEG responses $\mathbf{r}$, recorded at a sample rate of f_s. To quantify how the EEG responded to changes in stimulus feature $\mathbf{s}$ over a range of time lags $[\tau_{\min}, \tau_{\max}]$, we construct a forward model. We can use a regularized least squares approach, such as ridge regression, to quantify the forward model weights $\mathbf{w}_f$ as follows:

$$\mathbf{w}_f = (\mathbf{S}^\top\mathbf{S} + \lambda\mathbf{I})^{-1}\mathbf{S}^\top\mathbf{r}$$

where $\mathbf{S}$ is the design matrix containing the time-lagged stimulus features, $\mathbf{I}$ is the identity matrix and λ is the regularization parameter for controlling overfitting. In practice, we can implement a forward model in mTRF-Toolbox by setting the direction parameter `Dir` to `1` as follows:

```
model = mTRFtrain(stim,resp,fs,Dir,tmin,tmax,lambda);
```

The function returns a structure containing the model weights (`model.w`), the corresponding time lags (`model.t`) and other model parameters. The design matrix is automatically generated by `mTRFtrain()` based on the values of `tmin` and `tmax` (in milliseconds) and `fs` (in Hertz). Note, the value chosen for `lambda` should be validated empirically beforehand using an appropriate cross-validation procedure (for further details, see *Model training and testing*).

To construct an analogous backward model $\mathbf{w}_b$, we rearrange the equation and apply the time lags to the EEG responses instead to produce the design matrix $\mathbf{R}$ as follows:

$$\mathbf{w}_b = (\mathbf{R}^\top\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{R}^\top\mathbf{s}$$

Backward models are implemented in mTRF-Toolbox using the exact same line of code as above, but this time by setting the direction parameter `Dir` to `-1`. When the backward direction is specified, `mTRFtrain()` automatically rearranges the equation as above and reverses the time lags to be $[-\tau_{\max}, -\tau_{\min}]$ so that the user does not have to recalculate any of the parameters manually.

As mentioned earlier, the regularization process can act as a low-pass filter, suppressing fast oscillatory components (i.e., noise) in the model. A variant of ridge regression, known as Tikhonov regularization, does this very well and can be implemented in `mTRFtrain()` by setting the `'method'` parameter to `'Tikhonov'`.

Additionally, we can implement a series of single-lag models (i.e., models that only map between single time points) by setting the `'type'` parameter to `'single'`. If `tmin` and `tmax` are a range of values (e.g., `[0, 400]`), then `mTRFtrain()` will return a set of models that map between every single time lag within that range.

Model training and testing

Once the design is in place, the model can be trained and tested on the stimulus-response data. Standard training procedures typically allocate 70–90% of the data to training (training set) and split the rest of the data between validation and testing (validation set and test set). This avoids training and testing on the same data, as this would cause the model to overfit to noise in the dataset, producing a poor estimate of model performance. The validation set is used for tuning the model hyperparameters, while the test data is held out until the end and used to evaluate the final model.

Model training. When training a model, it is crucial that we validate how well our model generalizes at predicting new data. For reliability, it is advisable to obtain multiple measures of model performance using a method such as cross-validation; this is a procedure whereby the training and validation sets are rotated throughout the dataset, allowing us to evaluate the model on every segment of data (except the held-out test set). This is particularly relevant for neural data such as EEG (which are prone to numerous types of sparse artifacts), as we wish to avoid validating our model on a bad segment of data that is not reflective of the entire dataset. If you are working with contiguous time-series data (e.g., EEG), we recommend partitioning the data into shorter contiguous segments for iterative testing. If the data are already split into multiple segments or trials (e.g., 20 × 1-minute trials), it can be easier to retain this partitioning and perform a leave-one-out cross validation. Note, discontinuous segments or trials of data should not be concatenated as data at the discontinuities at trial boundaries will introduce noise into our model.

Typically, cross-validation is performed on individual subject data, resulting in a subject-dependent model, but subject-independent models are also useful in the absence of a sufficient amount of data per subject (see *Model design*). If there are multiple experimental conditions, models must be trained separately on data within each condition, irrespective of whether a subject-dependent or independent approach is taken. For implementation of data partitioning and cross-validation in mTRF-Toolbox, see Box 2.

Regularization. It is important to ensure that our model does not overfit to the noise in our training data, especially when there is a limited amount of data or the model has a large number of parameters. We do this by employing a technique called regularization. Regularization of linear models can be achieved in a number of different ways, most of which converge on a similar model solution and yield similar model performance (see Wong et al., 2018). A common method known as ridge regression uses a free parameter called the ridge parameter to control the correlations between the weights in the model. This smooths the model weights by penalizing weights with large values (i.e., the square of the weights), reducing the variance and producing a model that generalizes better. A variant of ridge regression, known as Tikhonov regularization, imposes a constraint on the first derivative of the model weights which provides temporal smoothing and dampens fast oscillatory components in the solution (Lalor et al., 2006; Crosse et al., 2016a). However, this approach may cause cross-channel leakage for multivariate input features (e.g., spectrograms, phonemes etc.) in which case it may be better to use ridge regression. Both methods are provided as options in mTRF-Toolbox. Another common method known as LASSO and its variants controls the number of weights with non-zero values. In practice, regularization takes care of two important issues during model fitting: firstly, if a model has many parameters relative to observations and is thus susceptible to overfitting, it reduces the amount of non-zero weights or their absolute value to produce a model that generalizes better; secondly, neighboring datapoints within both the stimulus and EEG can be correlated along multiple dimensions (e.g., space, time, frequency), also known as collinearity, leading to spuriously noisy model weights as the model fitting attempts to adjust for highly correlated neighboring samples (for an example, see Fig. 7E in Crosse et al., 2016a).

Model testing. To evaluate model performance, the model is first used to predict a set of output features from a held-out set of input features (i.e., the test set). For a forward model, we predict a set of unseen neural responses and for a backward model we reconstruct a set of unseen stimulus features. For subject-dependent models, the test set is typically a single segment or trial of data from that same subject. For subject-independent models, the test set can be either 1) multiple segments/trials of data from some held-out subject or 2) an individual segment/trial of data from a subject whose remaining trials are included in the training set (see Subject dependency section in *Model design*). The latter approach will likely yield better performance due to the inclusion of subject-specific data in the model.

We then evaluate model performance by calculating a predictive score based on the similarity (or error) between our prediction and the original signal (i.e., ground truth). It is common to use Pearson's correlation coefficient as a measure of similarity because it quantifies how linearly close the dynamics of the prediction are to the ground truth, irrespective of the magnitude or mean of the signal being predicted. In the event that the relationship between the predicted and original signals is not linear, a Spearman's correlation can instead be used. It is also common to measure the error (i.e., the absolute distance) between the predicted and original signals. Standard error metrics include the mean squared error or mean absolute error which, unlike correlations measures, rely on the absolute magnitude of the signals. For implementation of model testing in mTRF-Toolbox, see Box 2.

Example experiment

We start by concatenating all our spectrogram frequencies, **S**, and all our phonetic features, **F** into a joint encoding model, **FS**. We then map this **FS** speech representation to each EEG channel (i.e., we fit a TRF model) using a leave-one-out approach (each trial contained one minute of an audiobook, and there were breaks in between trials). This will allow us to see how well each TRF can predict the data in the left-out trial. We can iterate through all trials as test trials to get a sense of the model's validity across all of our data. For each iteration, we use ridge regression with leave-one-out cross-validation, using the trials selected for training to identify the optimal ridge parameter. We carry out this procedure separately for any feature models we intend to evaluate and compare (**F**, **S**, and **FS**). In our example experiment, in the absence of any strong hypothesis about lateralization (Hickok and Poeppel, 2007), we average prediction accuracies across EEG channels in order to evaluate the performance of the models.

Further considerations

Memory usage issues might occur if you are working with very large segmentations of the data or long trials. If the trials used are too long (e.g., five-minute trials), you can use the `'split'` parameter in `mTRFcrossval()` to chop them into smaller segments in order to reduce memory usage (see also the `'fast'` parameter for efficient memory usage). Because of the efficient cross-validation procedure employed by `mTRFcrossval()`, increasing the number of folds does not greatly increase computation time[5].

In addition, the researcher can specify a smaller correlation window size (e.g., 20 seconds) to obtain multiple measures per fold. Note, reducing the window size below a certain threshold (typically <10 s) will begin to reduce the average correlation value and also cause spuriously high and low estimates. However, this feature may be useful for researchers interested in real-time decoding applications such as brain-computer interfaces (BCIs), and such spurious correlation estimates can be managed over time by using Bayesian filtering techniques such as fixed-lag state-space models (Miran et al., 2018).

[5] This is achieved by summing the covariance matrices across all trials only once at the beginning, storing the trial-specific covariance matrices in memory and iteratively subtracting each of them from the sum on every fold.

Box 2. Model training and testing implementation.

Suppose we want to train a backward model with time lags $[\tau_{\min}, \tau_{\max}]$ to reconstruct particular stimulus features from EEG responses recorded at a sample rate of f_s. We must first partition our stimulus and response data into separate training and test sets. In mTRF-Toolbox, the user can partition continuous data into any number of folds and specify one of these folds to be allocated for testing. For example, we can create 10 folds (9 training, 1 test) and specify fold 5 as our test set as follows:

```
[sTrain,rTrain,sTest,rTest] = mTRFpartition(stim,resp,10,5);
```

where `stim` and `resp` are matrices of continuous data from a single subject. The output variables `sTrain` and `rTrain` are returned as 9-by-1 cell arrays containing the training set, and `sTest` and `rTest` are matrices containing the test set. Note, if the data were recorded as separate trials and are already stored as cell arrays, this step can be skipped.

We then conduct a cross-validation procedure on the training set to identify the optimal value for the regularization parameter λ as follows:

```
cv = mTRFcrossval(sTrain,rTrain,fs,Dir,tmin,tmax,lambda);
```

The function returns a structure containing the cross-validation statistics such as the correlation coefficient (`cv.r`) and the mean squared error (`cv.err`). Here, the regularization parameter `lambda` is a vector of values (e.g., `10.^(-6:2:6)`) over which we cross-validate our model. We set the direction argument `Dir` to `-1` to implement a backward mapping. To determine the optimal λ value, we average the performance metrics across folds, and take the λ corresponding to the maximum correlation value (or minimum error value):

```
[rmax,idx] = max(mean(cv.r));
```

where `rmax` is the maximum correlation and `idx` is the index of the λ value that yielded `rmax`. Note, if the stimulus features are multivariate, then the researcher will have to decide how to consolidate the data, e.g., take the mean or max across features (see also the banded ridge method in *Comparing different stimulus features*). We then use this lambda value to train our final model as follows:

```
model = mTRFtrain(sTrain,rTrain,fs,Dir,tmin,tmax,lambda(idx));
```

where `model` is a structure containing the relevant model parameters (see Box 1). We then test our model on the held-out test set as follows:

```
[pred,stats] = mTRFpredict(sTest,rTest,model);
```

where `pred` is a matrix of the predicted stimulus features (or EEG responses for a forward model) and `stats` is a structure containing the test statistics (`stats.r`, `stats.err`).

Evaluating model integrity

While the use of encoding/decoding models is well established in cognitive neuroscience and has been shown to reliably quantify different sensory and cognitive processes, it is still paramount that we assess the integrity of each individual participant's model before proceeding with our analysis. There are numerous factors that could lead to the construction of a model that does not reflect a meaningful stimulus-response relationship: 1) excessive noise in the neural data, caused by movement or external interference (see *EEG preprocessing*), such that it overwhelms the neural response to the stimulus, 2) poor subject compliance due to lack of motivation or fatigue, resulting in the subject perceiving little or no information, 3) certain subjects can exhibit inherently weak responses on the scalp to certain stimuli due to various anatomical reasons. Thus, it is critical to establish from the outset that a given participants' model is meaningful and performing well above chance level.

Establishing data integrity. We have previously described our plan to construct a forward model in order to quantify the neural tracking of acoustic and phonetic features. But is the SNR of our EEG data good enough to address such research questions? Before constructing our forward model, and as a first pass to assess whether the EEG responded to changes in the stimulus, we recommend first using a backward model to reconstruct a continuously time-varying stimulus feature that is known to be reliably tracked by the EEG signal. For example, we suggest using the speech envelope of the audio recording as it serves as a good proxy for the stimulus intensity. The reason we recommend this initial backward modelling step is that it is better able to detect the presence of neural tracking of the stimulus than forward modelling; as previously mentioned, backward modelling has the distinct advantage of utilizing all the available EEG channels simultaneously, as well as projecting to the stimulus domain where we typically have direct access to the ground truth (unlike noise-ridden EEG). If the accuracy of the resulting reconstruction is above chance, then we can then claim that the EEG contains genuine information relating to the desired stimulus feature.

Defining the null distribution. Quantifying chance-level performance of encoding/decoding models is non-trivial. For any statistical test, defining the null distribution makes inherent assumptions about the distribution of the data. Thus far, we have suggested validating the model prediction using Pearson's correlation coefficient, but unfortunately the standard statistical tests for correlation are not appropriate to determine if neural tracking is above chance. This is because they usually assume independent samples, but EEG data and naturalistic stimuli will typically exhibit a high correlation between neighboring time points or channels/features meaning adjacent samples are not independent. Instead, we recommend computing a null distribution of prediction accuracies using randomly shuffled permutations of the data at hand (Nichols and Holmes, 2002; Combrisson and Jerbi, 2015). This process involves random pairings of EEG responses to stimulus representations (for example, by shuffling the trial label). Note, however, that significance testing using this null distribution is only as sensitive as the number of trials. For example, 10 trials will produce at most 100 pairings, and randomly selecting 1000 pairings will produce several repeated pairings.

Alternatively, if the trials are sufficiently long or if there were not enough trials to get a null distribution of random pairings, one can (randomly) circularly time shift the stimulus relative to the reconstruction. This maintains the temporally correlated structure of both the stimulus and the reconstruction, while eliminating the phase relationship between the stimulus and reconstruction (Bialek and de Ruyter van Steveninck, 2005). One downside of this approach is that it does not account for time-locked responses that are present in all trials (such as an evoked response at the start of the trial) and could produce a false positive if the uniqueness of the reconstruction is of interest. Additionally, discontinuities introduced by circular shifting (if the start and end points of the signal are very different) can produce inappropriate null distributions (Harris, 2020), but this issue is somewhat minimized by high-pass filtering the data (see *EEG preprocessing*). The experimenter should decide on the most appropriate method for creating a null distribution, based on the experiment design and the number and duration of the stimuli.

Quantifying significance. Once a null distribution has been defined, we can then calculate a *p*-value for an individual participant's model by averaging their performance measures (e.g., correlation coefficients) across trials and quantifying the proportion of the null distribution that falls above their average performance value (i.e., a one-tailed test). Alternatively, to account for the variability in true correlations, we can compute a measure of sensitivity known as d-prime (*d'*) which is computed as follows (see also Figure 3):

$$d' = \frac{\mu_{\text{true}} - \mu_{\text{null}}}{\sqrt{\frac{1}{2}\left({\sigma^2}_{\text{true}} + {\sigma^2}_{\text{null}}\right)}}$$

To calculate significance for a group of participants, we take the average of each participant's null distribution and compare that sample of null correlations to a sample of true correlations using an appropriate means test.

Example experiment

Before testing for neural differences between our dyslexia and control groups, we wish to verify the general integrity of our data, i.e., whether we can reliably detect neural responses to the stimulus in the EEG data. We use backward modeling to derive a mapping from the 128-channel EEG data to the speech envelope for each trial. Using the same time-lags we selected for our forward model (0–300 ms), we conduct a leave-one-out cross-validation to compute the average reconstruction accuracy. We assess the significance of our model's performance against a null distribution derived from random permutations of the data. Because our observed reconstruction accuracy is above the 95th percentile of the permuted reconstruction accuracies, we conclude that the stimulus can be decoded better than chance and move forward with our main analysis.

Next, we've chosen to evaluate our **F**, **S**, and **FS** models averaged across all EEG electrodes (Figure 1). We evaluate the models by averaging their prediction accuracies and perform the same permutation procedure on that average, randomly shuffling trials to generate a null distribution. Because all three models perform better than chance, we conclude that acoustic and phonetic features were reliably encoded in the EEG data. However, we know that acoustic and phonetic features are highly correlated in speech. Next, we will disentangle the relative contributions of these different stimulus features.

Further consideration

In addition to running an initial backward model analysis to assess data integrity, one could also include a short ERP study before the main experiment. This could consist of very simple stimuli such as flashes or beeps, depending on the modality of interest. While not a direct measure of continuous stimulus tracking, it would give the experimenter some idea of the quality of their data independent of the modelling analysis which, when performed incorrectly, can produce poor results despite the data potentially being of good quality. Another advantage to this approach is that there are ways to rapidly analyze ERPs online and establish data quality early on in the experiment. This can help in deciding whether to continue running the entire study on a given participant or terminating it early to save you both time.

Box 3. Model evaluation implementation.

Regression models are typically evaluated using correlation or error metrics. In mTRF-Toolbox, we can compute such metrics for a forward or backward model using the ground truth `y` and the model prediction `pred` as follows:

```
[r,err] = mTRFevaluate(y,pred);
```

where `r` is the Pearson correlation coefficient and `err` is the mean squared error (MSE). Alternatively, we can specify a Spearman correlation using the `'corr'` input argument and the mean absolute error (MAE) using the `'error'` argument. The `mTRFevaluate()` function is automatically called by other functions in mTRF-Toolbox such as `mTRFpredict()` and `mTRFcrossval()` for computing such performance metrics.

In order to evaluate whether such performance metrics are statistically significant, we can use a permutation-based approach which cross-validates models using mismatched permutations of the stimulus features and neural responses. In mTRF-Toolbox, this can be implemented as follows:

```
stats = mTRFpermute(stim,resp,fs,Dir,tmin,tmax,lambda);
```

The function returns a structure containing the usual cross-validation statistics such as the correlation coefficient (`stats.r`) and the mean squared error (`stats.err`). An optional input argument `'nperm'` can be used to specify the number of permutations to perform.

Comparing different stimulus features

One of the strengths of using linear models is the ability to define multiple stimulus representations, such that we can target specific stages of processing along the sensory/cognitive pathway. While we can directly compare the performance of various models constructed using the different stimulus features, it is likely that such features contain overlapping information along one or more dimensions leading to redundancy between the corresponding models. However, it may be necessary to quantify the unique contribution of a specific feature if the experimenter is interested in obtaining a measure of something like speech-specific processing. There are multiple ways of approaching this problem.

Combining stimulus features. This approach first quantifies the combined contribution of multiple features by concatenating them together along the columns of the design matrix and constructing a "combined" model (see Fig. 1A; Di Liberto et al., 2015; O'Sullivan et al., 2017; Desai et al., 2021). To evaluate whether an individual stimulus feature contributes unique information, separate models are fit using the individual features and the difference in prediction accuracy between the combined model and the individual models is computed. While this is an indirect way to quantify such unique contributions, it has been shown to be predictive of behavior (Di Liberto et al., 2018a). Note, when combining multiple stimulus features in a single matrix, regularization may not affect the features equally, especially if the frequency content differs between the features (see *Further Considerations* below). In general, this method is more suited to forward modelling because it is easier to compare the performance of different stimulus feature models in the EEG domain than in the stimulus domain; certain stimulus features may consist of features which are difficult to reconstruct (e.g., sparse binary representations), requiring additional non-linear transformations.

Partialling out contributions. A more direct way to deal with common brain activations produced by redundant stimulus features is to partial them out of the EEG data (O'Sullivan et al., 2021). The residual EEG signal can then be mapped in a forward or backward direction to the other stimulus features (i.e., the features that were not regressed out of the EEG) in order to define a model that captures the neural dynamics unique to those features. Vice versa, the same procedure can then be implemented for the other set of features. Depending on how much overlap there is between the two sets of features (temporal, spectral etc.), the residual EEG may contain considerably less information about the other stimulus set after the first has been partialled out, resulting in very low prediction accuracy scores. However, the absolute value of the prediction accuracy is not often not of great importance; for example, when comparing different subject groups or experimental conditions, we are often more interested in the prediction accuracy score relative to that of another group or condition.

Example experiment

Our original hypothesis was that there should be group differences in phonological but not acoustic processing. We have already determined how well we can predict data using a speech representation that includes the combined acoustic and phonetic features, i.e., **FS** (see *Example experiment* section in *Evaluating model integrity*). Following that, we aim to predict the data on the same channels using mTRF models fit to the spectrogram (**S**) representation alone. Then, by subtracting those **S** predictions from the **FS** predictions, we can determine by how much the EEG predictions were improved when including phonetic (**F**) features along with the spectrogram (**S**). Conversely, we can isolate a measure of acoustic processing by seeing how much the **FS** representation outperforms an **F** representation alone. In our experiment, we are particularly interested in testing the hypothesis that **FS**–**F** (quantifying acoustic processing) is similar for both groups, but that **FS**–**S** (quantifying phoneme processing) is larger for our control group than the dyslexia group (Fig. 2A). This pattern of results would be consistent with our hypothesis that there is a specific impairment in phonological processing in our clinical group. We do this, being careful to regularize each model separately for each subject to optimize its performance using all EEG channels.

Further considerations

In the original study from which our example experiment is based (Di Liberto et al., 2018b), some additional analyses were based on subject-independent models, calculated based on an average across subjects. For each group, data from all subjects but one was combined to derive a single TRF and used to predict the EEG signal of the held-out participant. Figure 3 shows the EEG prediction correlations of the subject-independent models fit for **S**, **F**, and **FS**. The main result is that the control (CTR) and dyslexia (DX) groups showed different spatial patterns in prediction accuracies. While there was an overall reduction of that metric in dyslexia, the correlations also increased over some scalp areas, pointing to an atypical processing rather than simply reduced processing or increased inter-subject variability. Furthermore, the atypical processing was most pronounced for **FS**, with further analyses (not shown) revealing a correlation between **FS**–**S** and behavioral scores of phonological awareness and memory.

When estimating multiple-feature models, such as our **FS** model described above, with ordinary regularization methods (e.g., ridge), the use of a single regularization parameter may lead to sub-optimal model fits and poor predictions. This is because they assume the same level of regularization across all feature types (e.g., spectrogram, phonetic features). The amount of regularization required for each feature can vary depending on several factors such as frequency content, magnitude of the stimulus vector, stimulus sparsity, and EEG SNR. In our **FS** model, we have two markedly different types of speech features; the spectrogram is represented as a continuous value across 32 frequency bands while phonetic features are discrete, binary variables across 19 different phonetic features. As such, in our example, we can fit a combined encoding model while applying separate levels of regularization to each feature type (known as *banded ridge regression*): one “band” for the spectrogram and one for the phonetic features (see Box 4). This approach has the added benefit of reducing spurious correlations between features. Fit separately, single-feature models may produce prediction accuracies that are over-estimated due to correlations between the features. Banded ridge regularization can reduce this problem by “decorrelating the model features to an amount determined by their covariance and the regularization parameters” (Nunez-Elizalde et al., 2019). We can fit a combined model (**FS**) with banded regularization parameters and then evaluate the prediction accuracy of each individual feature (**F** or **S**) separately using the model weights estimated by the combined model (for more detail see Nunez-Elizalde et al., 2019).

Box 4. Comparison of stimulus features implementation.

In order to compare between the unique contributions of different stimulus features to the observed brain activity, we must account for statistical and perceptual redundancies between those features. As outlined in this section, one way to do this is to compute the difference in performance between combined and individual feature models. A more direct method involves partialling out the contribution of one set of features from the EEG data, then modelling the relationship between another set of features and the residual EEG. In mTRF-Toolbox, we can implement this partialling procedure and obtain the residual EEG responses as follows:

```
resid = mTRFpartial(stim,resp,fs,Dir,tmin,tmax,lambda);
```

As discussed in this section, we have adapted banded ridge regression as described in Nunez-Elizalde et al. (2019). In short, this function performs leave-one-out cross-validation while applying different levels of regularization separately to "bands" of stimulus features. Suppose we wish to apply 2 bands of regularization to a stimulus representation that has 10 features, with 5 features per band. First, we define the range of regularization values for each band and how we wish to group the features in each band:

```
band1 = 10.^(-6:2:6);
band2 = 10.^(0:2:12);
lambda = [band1;band2];
grouping = [1,1,1,1,1,2,2,2,2,2];
```

Then we perform our banded regularization in mTRF-Toolbox as follows:

```
cv = mTRFcvbanded(stim,resp,fs,Dir,tmin,tmax,lambda,grouping);
```

The function returns a structure containing the usual cross-validation statistics. It is important to consider that the computation time grows exponentially with the number of bands in an exhaustive search (${N_{\lambda}}^{N_{bands}}$). However, the cross-validation procedure can be optimized via the search algorithm in MATLAB, which is implemented in the following mTRF function:

```
cv = mTRFcvsearch(stim,resp,fs,Dir,tmin,tmax,grouping);
```

Note that lambda is no longer a required input. Instead, it is an optional parameter, `'init'`, which specifies an initial lambda value for each band (default value is 1). Additionally, if `'grouping'` is omitted, the function performs the standard (single-band) cross-validation described in Box 2 via the search algorithm.

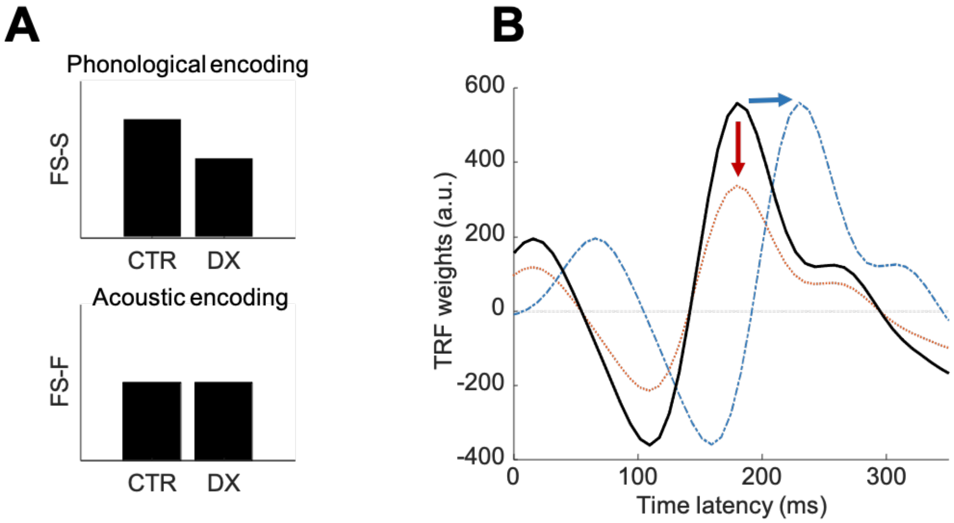

FIGURE 2 | Graphical depiction of expected results for example clinical study using forward model approach. (A) We hypothesize that there is a deficit in the neural processing of phonetic features in the dyslexia group (DX) relative to our control group (CTR). By looking at the differential in prediction score (Pearson's *r*) between the combined model (spectrogram and phonetic features, FS) and the spectrogram alone (S), we would expect to see a larger effect in the control group, indicating a larger contribution of phonetic features to the FS model performance in controls. In contrast, when we compare FS to a model with only phonetic features (F), there should be no significant difference in the difference in prediction accuracies. **(B)** Hypothetical group differences in average TRF weights. The average TRF weights for the control group is shown as the black trace. One possible scenario shows a reduction in TRF amplitude for the dyslexia group (red trace), which could indicate either reduced neural activity or increased inter-subject variability. Alternatively, there could be a difference in TRF latency (blue trace), due to delayed neural processing. This specific effect has been observed in a previous study quantifying semantic processing in older adults (Broderick et al., 2021).

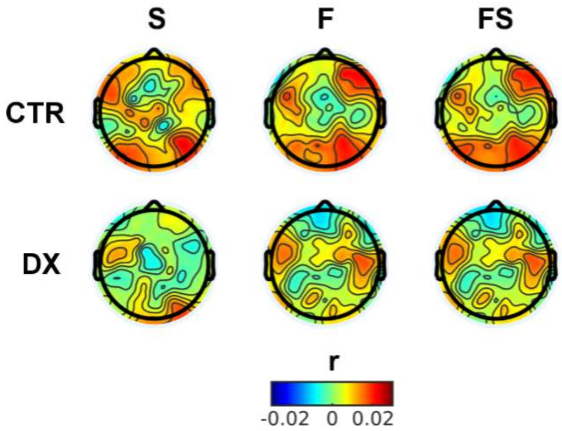

FIGURE 3 | Group differences in distribution of forward model prediction scores. Scalp topographies of prediction scores (Pearson's *r*) for control (CTR) and dyslexia (DX) groups using subject-dependent forward models (adapted from Di Liberto et al., 2018b). These results show that there are differences in prediction score, as per our hypothesis, but also that there can be differences in the distributions of those scores across the scalp.

Interpreting the model weights

One of the major advantages of using linear models is that their weights are easy to interpret, a property that is highly desirable when studying physiological systems. In this section, we discuss interpretation in the context of forward and backward models, model generalization, as well as presenting simulations that demonstrate the impact of data quantity and signal-to-noise ratio (SNR) on prediction score.

Forward model. The analysis and interpretation of forward model weights are similar to that of an ERP, although a few caveats should be noted. Firstly, a forward model is not an ERP because its weights are fitted to optimally predict EEG data, thus, the relationship between the weights is mathematically relevant by design (Kriegeskorte and Douglas, 2019). Forward models trained on naturalistic stimuli produce convincing representations of neural responses with similarities to ERPs (Lalor et al., 2006; Lalor et al., 2009), but correlations between stimulus parameters can affect this interpretation. Secondly, if our goal is to capture the underlying neural response using linear modeling, we have to ensure that the model generalizes well at predicting new data. Both the model's approximation of the underlying response and its performance vary as a function of SNR and the amount of available data. As such, it is crucial to evaluate the model's predictive power prior to interpreting the model weights. It can be tempting to interpret specific temporal or spatial features of the model weights as reflecting something interesting about the underlying brain activity, particularly when they appear to satisfy our preconceived notions or hypotheses regarding brain function. However, if a model has no predictive power, then it is likely overfit to noise in the data.

Backward model. While it is possible to quantitatively compare patterns of backward model weights across multiple experimental conditions (e.g., Crosse et al., 2015), it is not recommended to visualize or interpret such patterns in a qualitative manner or through the lens of neurophysiology. As previously stated, the weights corresponding to backward models are not physiologically relevant because they map in the acausal direction, i.e., in reverse to the natural flow of information of the system under study. However, there are transformations that can be applied to backward models in order to observe the corresponding forward representation (see Haufe et al., 2014). In mTRF-Toolbox, the function `mTRFtransform()` allows the researcher to perform this transformation in one line of code (see Box 5). Note, we strongly recommend using the resulting transformed model merely for the purpose of interpretation, and not for predicting neural responses to novel stimuli. There is no guarantee that the resulting forward model from the transformation is sufficiently regularized and optimal for EEG prediction.

Simulations. To demonstrate this issue of model performance and interpretability as a function of SNR, we conducted a series of linear modeling analyses on simulated EEG data with a frequency range relevant to speech tracking (2–15 Hz, roughly theta to beta). For each simulation, we: 1) randomly generated a time-varying 'stimulus' in this frequency range and convolved it with the expected TRF, 2) added noise with approximately the same spectrum as EEG, and 3) trained and tested the model using a leave-one-trial-out procedure, where each trial was one minute long. Each simulation was run 100 times. Based on these simulations, we show that model performance improves with more data and when the data are less noisy (i.e., higher SNR), and thus the resulting model has a high correlation with the true response (Fig. 4A). As the amount of data or the SNR decreases, both the model performance and the model fit drop. Note that 'noise' in these simulations refers to any component that does not track the stimulus. Practically, this includes external mechanical and electrical noise as well as uncontrollable factors such as neural activity from other brain regions and processes that are not of interest.

Collapsing the results across SNR and data quantity, we see a direct relationship between the reliability of the model's representation of the true response and the prediction score (Fig. 4B). Here we use a d-prime measure, which is quantified relative to a null distribution produced by randomly circularly shifting the trials (see the section *Evaluating model integrity).* When the prediction accuracy is low, the model is a poor representation of the true response, and this increases with prediction accuracy. More specifically, the correlation between the predicted and true model plateaus for d-prime prediction accuracies around 2, and the model is often (> 90% of the time) a reliable representation of the true response (Figure 4B, see also Figure 4C, D, and E for example simulations).

Thus, we strongly recommend examining prediction accuracy before interpreting model weights. Even with an experimental setup with low electrical noise and well-behaved subjects, SNR can vary across subjects due to differences in cortical folding or the relative activity in other brain regions that are not of interest. Additionally, when the data are particularly noisy, it is possible to get a poor TRF estimate, even with a lot of data (Figure 4B, C). For this reason, we also recommend having at least 10 subjects in the experiment since TRF estimates can be somewhat variable, even at high SNRs.

Example experiment

So far, we have compared the control and dyslexia groups based on prediction accuracy (see *Model comparison for different stimulus features*). It is also possible that there will be differences in the evoked responses to the different features, which will be captured by the model weights themselves. Differences might manifest as a change in magnitude or a change in delay of the peaks and troughs in the TRF (Fig. 2B). For simplicity, our focus will be on a TRF representing one stimulus feature, but this approach can be extended to examine multiple stimulus features.

To study the TRF weights in our example, we carried out a two-step procedure. First, we identified time-lags with TRF weights that were significantly different from zero within each group. We used a Wilcoxon test with FDR-correction, but this analysis could also be conducted, for example, with a cluster mass statistics method (Maris and Oostenveld, 2007). Then, similar statistical methods were used to evaluate between-group effects. In our example experiment, we observe no between-group differences between any acoustic weights, but we observe between-group difference in the phonetic features' weights at time lags around 200 ms.

An alternative data-driven approach consists of performing a cluster analysis on the TRF weights to identify clusters with significant TRF components. While a single 2D cluster analysis (EEG channel x time-lag) is sufficient for univariate inputs, multiple 2D cluster analyses should be run when multiple speech features have been included in the model. The analysis can then be carried out for each cluster by averaging each set of electrodes or by focusing on the centroid channel of each cluster.

Box 5. Backward to forward model transformation.

Suppose we want to train a backward model with time lags $[\tau_{\min}, \tau_{\max}]$ to reconstruct particular stimulus features from EEG responses recorded at a sample rate of f_s. We would first optimize the regularization parameter λ using a cross-validation procedure as outlined in Box 2. Once complete, we can train our backward model in mTRF-Toolbox as follows:

```
bmodel = mTRFtrain(stim,resp,fs,Dir,tmin,tmax,lambda);
```

Aside from using our backward model for decoding purposes, we may wish to gain a deeper insight into the underlying neurophysiology of the related brain activity. While we cannot directly interpret the weights of our backward model in its current from, we can transform it into the corresponding forward model as described in Haufe et al. (2014) using the mTRF-Toolbox:

```
fmodel = mTRFtransform(bmodel,resp);
```

The function returns a structure `fmodel` containing the same model parameters as those returned by `mTRFtrain ()`, except for the bias term. Thus, the resulting model cannot (and should not) be used for prediction. For that, we recommend directly constructing an optimized forward model.

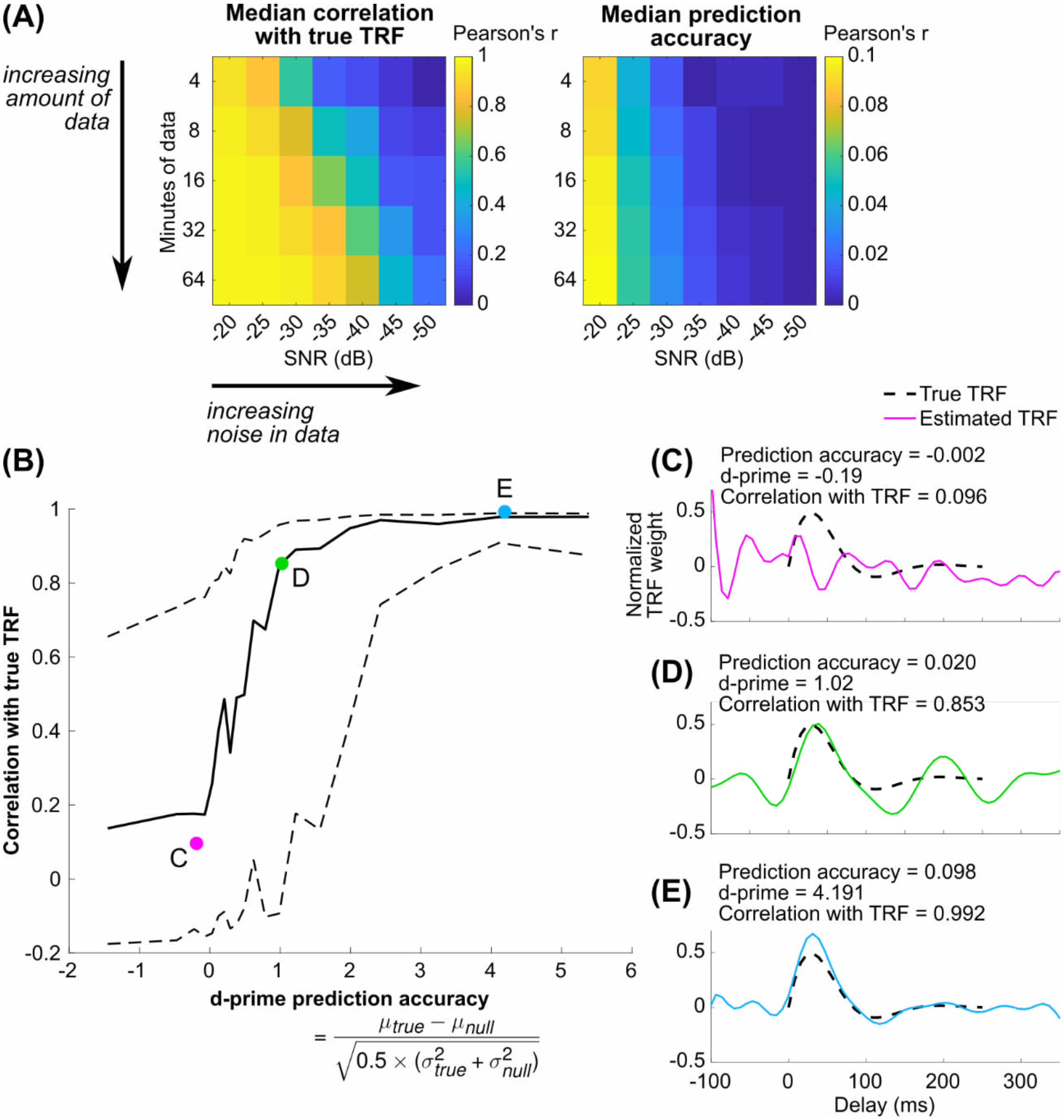

FIGURE 4 | We simulated data containing a TRF-based response with added EEG-shaped noise. Both were filtered between 2–15 Hz. Each simulation was re-run 100 times with new noise and a new randomly generated stimulus using different SNRs (ratio of response to noise root-mean-squared) and a different number of trials (each trial is 1 min long). **(A)** Leave-one-trial-out procedure was used to quantify prediction accuracy of the trials, and for each simulation we averaged prediction accuracies across trials. Shown are the median correlations between the true TRF and the optimal modeled TRF across simulations, and similarly shown are the median prediction accuracies across simulations. Both prediction accuracy and the model estimate of the true TRF decrease with increasing amount of noise and decreasing number of trials. In light of this, we collapsed the data across conditions plotted the relationship between prediction accuracy and model TRF to true TRF correlation across simulations **(B)**. d-prime prediction accuracy was used to normalize for differences in the null distribution, which can vary with the frequency range of the data. Shown for each condition are the median (solid line) and the 10– 90% quantiles (dashed lines). As prediction accuracy decreases, the model estimate of the true TRF gets less reliable. **(C, D, E)** Shown are example stimulations with poor, moderate, and good estimates of the TRF, respectively (C: −45 dB SNR, 64 minutes; D: −25 dB SNR, 4 minutes; E: −20 dB SNR, 64 minutes). The root-mean-square of the estimated TRFs were normalized in this plot to match the true TRF. The d-primes and correlations between the true and predicted model for each simulation have also been labeled in B using the same colors of the traces in C, D, and E.

What can go wrong?

Small Datasets and Regularization

The first step of fitting a model is identifying the optimal regularization parameter (*see Model training and testing*). A simple and common approach to identify the optimal regularization parameter(s) is to do an exhaustive search within a certain range of values. While an exhaustive search can be costly for multiple parameters, it can be feasible in the case of a single parameter, such as λ for ridge regression. The optimal λ value can be selected so that it optimizes the prediction score, which could be, for example, the EEG prediction correlation or the prediction MSE.

The optimal λ value may differ across subjects (or sessions), as it depends on factors such as the SNR. For this reason, the λ tuning is usually performed on individual subjects. However, this approach can be problematic when working with small datasets. Consider, for example, an EEG dataset with n = 20 participants, each with only 3 minutes of data. It is possible that such a small amount of data is not sufficient to fit generalized models that can reliably predict new EEG data. In other words, there may be particular λ values producing models that are better than others, but there may not be enough EEG data to reliably assess the quality of those models. This issue may prevent us from deriving reliable models at the individual-subject level with the abovementioned standard exhaustive search. One possible solution is to identify a suitable denoising procedure to increase the SNR, which would increase the reliability of the prediction score. If that is not possible or sufficient, it may be possible to improve the model fit by including additional assumptions. For instance, if we assume the EEG responses for individuals within the same group is similar, data from all participants can be combined into a single dataset to determine a unique regularization parameter. This could be implemented in different ways that depend on particular assumptions and the dataset (e.g., should the EEG data be normalized before it is combined between subjects? How are the data combined?). The researchers may then decide to use the resulting lambda to study the TRFs at the group level or to restrict the parameter search at the individual subject level.

Poor parameter selection

Consider the scenario where a researcher wants to assess auditory processing in a particular cohort of interest using a music listening paradigm. Before doing so, they plan to first validate their analysis pipeline on a publicly available dataset (Di Liberto et al., 2020). In this study, EEG data were recorded while 20 participants listened to ten monophonic piano melodies from Bach, each occurring three times, and presented in random order. The researcher aims to fit envelope TRFs on all participants and verify that they exhibit the typical characteristics of an auditory TRF that were shown in previous research.

The team decides to select the following parameters: a) The EEG frequency-band 1-30 Hz; b) The time-lag window [–100, 400], which should be sufficient to capture the expected responses between about 20 and 250ms; and c) The range of possible regularization values [1, 10, 10^2, 10^3, 10^4, 10^5]. As for our indications, ~~t~~he team first examined the EEG prediction correlation values (Figure 5A). The values were of the order of 0.025, so not particularly large, but significantly larger than zero. Thus, they decide to move to the next step and interpret the TRF weights (Figure 5B). The resulting TRF shows two slow "components": One arising before 0 ms lag and terminating around 200ms; The other is a sustained response with inverse polarity that starts at 200ms. This may appear as an interesting result, potentially reflecting the predictive nature or music perception. However, the response looks too smooth compared with previous results and they decide to double-check why that is the case. In doing so, they realize that the selected optimal lambda always had the minimum value in the search interval (Figure 5C). They then extended the search to a broader interval from 10^{-7} to 10^5 (logarithmically spaced), verifying that the optimal lambda was within that interval and not at its border for all subjects (Figure 5F). As a result, they found much larger EEG prediction correlations (~0.05; Figure 5D) and obtained TRFs that match their initial expectation (Figure 5E).

The team's first pass of optimizing the model (Figure 5A, B, C) is a classic example of poorly choosing the free parameter. The issue is not necessarily due to the choice of lambda values, but it is rather due to a missing initial check on the tuning curve. It is important to ensure that the tuning curve has reached a plateau before stopping the parameter search.

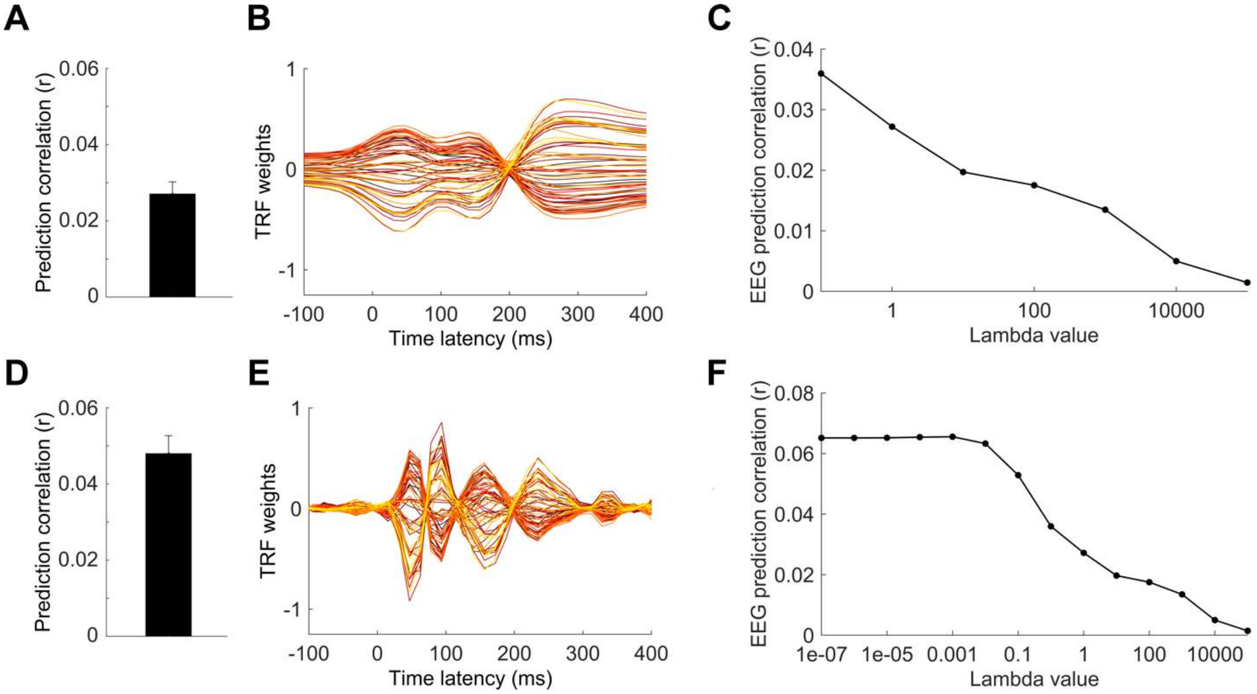

FIGURE 5 | Misinterpreting the TRF result because of poor regularization. (A) EEG prediction correlation, averaged across all channels, when the optimal lambda was determined within the interval [0.1, 10^5]. Error bars indicate the SEM across participants. **(B)** TRF weights of the corresponding model for individual electrodes. Darker colors indicate electrodes in a progressively more posterior position. **(C)** Tuning curve for the regularization parameter λ averaged across participants. The optimal lambda was the minimum value in the search interval 0.1. **(D, E, F)** Results when the lambda search was performed in the larger interval [10^{-7}, 10^5]. The optimal lambda was 10^{-3}.

Conclusion

Linear modeling allows researchers to both interpret and quantify the reliability of neural tracking to continuous stimuli. We have provided an example experiment throughout in which linear models might be used to study phonological processing in a subject group with dyslexia. This experiment also highlights how comparisons of EEG prediction accuracy between two models can be used to quantify the information provided by an additional feature relating to the speech, such as phonological content. The mTRF-toolbox provides all of the tools needed to do linear modeling and address questions of neural response reliability and comparisons between models. Further details can be found in a previous manuscript (Crosse et al., 2016a), but here our aim was to detail other steps of experiment design, preprocessing, and interpretation when linear modeling is used to test hypotheses in applied research.

Computational cognitive neuroscience is a rapidly advancing field. We have focused on linear modeling because of our experience with this technique and because it is a simple method with which one can produce interpretable models from continuous and naturalistic stimuli. But it is of course not the only method of neural data analysis. The relative utility of linear models versus nonlinear models (which assume a nonlinear relationship between stimulus properties and the neural response) is an active area of debate (Ivanova et al., 2020). Furthermore, deep neural networks have shown promise for generating artificial neural responses that mimic the processing stages of sensory systems in the brain (Yamins and DiCarlo, 2016; Richards et al., 2019). Depending upon the research questions, we think more complex, nonlinear models can also be useful for clinical research, and as such they are worth paying attention to in the near future. Still, linear models provide a more direct bridge between the controlled experiment design of previous work and machine-learning-based analyses that work for experiments with continuous stimuli and multiple time-varying features of interest (Holdgraf et al., 2017). Because of this, linear models will remain important as the field of cognitive neuroscience continues to advance, and we look forward to these advances.

Code availability

The latest code for the mTRF-Toolbox, including sample demos and data, can be found on GitHub (https://github.com/mickcrosse/mTRF-Toolbox). The code for the simulations used to create Figure 4 can also be found on GitHub (https://github.com/natezuk/mTRF-Toolbox-simulations).

Acknowledgements

We would like to thank Andrew Anderson for his comments and suggestions on this manuscript.

References

Bialek, W., and De Ruyter Van Steveninck, R.R. (2005). Features and dimensions: Motion estimation in fly vision. *arXiv preprint q-bio/0505003*.

Brodbeck, C., Hong, L.E., and Simon, J.Z. (2018). Rapid transformation from auditory to linguistic representations of continuous speech. *Current Biology* 28**,** 3976-3983. e3975.

Broderick, M.P., Di Liberto, G.M., Anderson, A.J., Rofes, A., and Lalor, E.C. (2021). Dissociable electrophysiological measures of natural language processing reveal differences in speech comprehension strategy in healthy ageing. *Scientific reports* 11**,** 1-12.

Carandini, M., Demb, J.B., Mante, V., Tolhurst, D.J., Dan, Y., Olshausen, B.A., Gallant, J.L., and Rust, N.C. (2005). Do we know what the early visual system does? *Journal of Neuroscience* 25**,** 10577-10597.

Combrisson, E., and Jerbi, K. (2015). Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of neuroscience methods* 250**,** 126-136.

Crosse, M.J., Butler, J.S., and Lalor, E.C. (2015). Congruent Visual Speech Enhances Cortical Entrainment to Continuous Auditory Speech in Noise-Free Conditions. *The Journal of Neuroscience* 35**,** 14195-14204.

Crosse, M.J., Di Liberto, G.M., Bednar, A., and Lalor, E.C. (2016a). The multivariate temporal response function (mTRF) toolbox: a MATLAB toolbox for relating neural signals to continuous stimuli. *Frontiers in human neuroscience* 10.

Crosse, M.J., Di Liberto, G.M., and Lalor, E.C. (2016b). Eye can hear clearly now: inverse effectiveness in natural audiovisual speech processing relies on long-term crossmodal temporal integration. *Journal of Neuroscience* 36**,** 9888-9895.

De Cheveigné, A., Di Liberto, G.M., Arzounian, D., Wong, D.D., Hjortkjaer, J., Fuglsang, S., and Parra, L.C. (2019). Multiway canonical correlation analysis of brain data. *NeuroImage* 186**,** 728-740.

De Cheveigné, A., and Nelken, I. (2019). Filters: when, why, and how (not) to use them. *Neuron* 102**,** 280-293.

De Cheveigné, A., and Parra, L.C. (2014). Joint decorrelation, a versatile tool for multichannel data analysis. *Neuroimage* 98**,** 487-505.

De Cheveigné, A., Wong, D.D., Di Liberto, G.M., Hjortkjaer, J., Slaney, M., and Lalor, E. (2018). Decoding the auditory brain with canonical component analysis. *NeuroImage* 172**,** 206-216.

Desai, M., Holder, J., Villarreal, C., Clark, N., and Hamilton, L.S. (2021). Generalizable EEG encoding models with naturalistic audiovisual stimuli. *bioRxiv*.

Di Liberto, G.M., Crosse, M.J., and Lalor, E.C. (2018a). Cortical Measures of Phoneme-Level Speech Encoding Correlate with the Perceived Clarity of Natural Speech. *eNeuro* 5.

Di Liberto, G.M., and Lalor, E.C. (2017). Indexing cortical entrainment to natural speech at the phonemic level: Methodological considerations for applied research. *Hearing research* 348**,** 70-77.

Di Liberto, G.M., O'sullivan, J.A., and Lalor, E.C. (2015). Low-Frequency Cortical Entrainment to Speech Reflects Phoneme-Level Processing. *Current Biology* 25**,** 2457-2465.

Di Liberto, G.M., Pelofi, C., Bianco, R., Patel, P., Mehta, A.D., Herrero, J.L., De Cheveigné, A., Shamma, S., and Mesgarani, N. (2020). Cortical encoding of melodic expectations in human temporal cortex. *Elife* 9**,** e51784.

Di Liberto, G.M., Peter, V., Kalashnikova, M., Goswami, U., Burnham, D., and Lalor, E.C. (2018b). Atypical cortical entrainment to speech in the right hemisphere underpins phonemic deficits in dyslexia. *NeuroImage* 175**,** 70-79.

Ding, N., Chatterjee, M., and Simon, J.Z. (2014). Robust cortical entrainment to the speech envelope relies on the spectro-temporal fine structure. *Neuroimage* 88**,** 41-46.

Ding, N., and Simon, J.Z. (2012). Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *Journal of neurophysiology* 107**,** 78-89.

Ding, N., and Simon, J.Z. (2013). Adaptive Temporal Encoding Leads to a Background-Insensitive Cortical Representation of Speech. *Journal of Neuroscience* 33**,** 5728-5735.

Ding, N., and Simon, J.Z. (2014). Cortical entrainment to continuous speech: functional roles and interpretations. *Frontiers in human neuroscience* 8.

Frey, H.P., Molholm, S., Lalor, E.C., Russo, N.N., and Foxe, J.J. (2013). Atypical cortical representation of peripheral visual space in children with an autism spectrum disorder. *European Journal of Neuroscience* 38**,** 2125-2138.

Giraud, A.-L., and Poeppel, D. (2012). Cortical oscillations and speech processing: emerging computational principles and operations. *Nature neuroscience* 15**,** 511-517.

Harris, K.D. (2020). Nonsense correlations in neuroscience. *bioRxiv*.

Harte, N., and Gillen, E. (2015). TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia* 17**,** 603-615.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage* 87**,** 96-110.

Hickok, G., and Poeppel, D. (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience* 8**,** 393-402.

Holdgraf, C.R., Rieger, J.W., Micheli, C., Martin, S., Knight, R.T., and Theunissen, F.E. (2017). Encoding and Decoding Models in Cognitive Electrophysiology. *Frontiers in Systems Neuroscience* 11**,** 61.

Huth, A.G., Lee, T., Nishimoto, S., Bilenko, N.Y., Vu, A.T., and Gallant, J.L. (2016). Decoding the semantic content of natural movies from human brain activity. *Frontiers in systems neuroscience* 10**,** 81.

Hyvärinen, A., and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks* 13**,** 411-430.

Irino, T., and Patterson, R.D. (2006). A dynamic compressive gammachirp auditory filterbank. *Audio, Speech, and Language Processing, IEEE Transactions on* 14**,** 2222-2232.

Ivanova, A., Schrimpf, M., Isik, L., Anzellotti, S., Zaslavsky, N., and Fedorenko, E. (2020). Is it that simple? The use of linear models in cognitive neuroscience.

Kriegeskorte, N., and Douglas, P.K. (2019). Interpreting encoding and decoding models. *Current opinion in neurobiology* 55**,** 167-179.

Lalor, E.C., and Foxe, J.J. (2010). Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution. *European Journal of Neuroscience* 31**,** 189-193.

Lalor, E.C., Pearlmutter, B.A., Reilly, R.B., Mcdarby, G., and Foxe, J.J. (2006). The VESPA: a method for the rapid estimation of a visual evoked potential. *Neuroimage* 32**,** 1549-1561.

Lalor, E.C., Power, A.J., Reilly, R.B., and Foxe, J.J. (2009). Resolving precise temporal processing properties of the auditory system using continuous stimuli. *Journal of Neurophysiology* 102**,** 349-359.

Mahajan, Y., Crosse, M.J., Ching, A., Borg, S., Kim, J., and Davis, C. (in prep). Test-retest reliability of encoding and decoding models of natural speech and its sensitivity to selective attention in EEG.

Maris, E., and Oostenveld, R. (2007). Nonparametric statistical testing of EEG-and MEG-data. *Journal of neuroscience methods* 164**,** 177-190.

Mesgarani, N., David, S.V., Fritz, J.B., and Shamma, S.A. (2009). Influence of context and behavior on stimulus reconstruction from neural activity in primary auditory cortex. *Journal of Neurophysiology* 102**,** 3329-3339.

Miran, S., Akram, S., Sheikhattar, A., Simon, J.Z., Zhang, T., and Babadi, B. (2018). Real-time tracking of selective auditory attention from M/EEG: A bayesian filtering approach. *Frontiers in neuroscience* 12**,** 262.

Murray, M.M., Brunet, D., and Michel, C.M. (2008). Topographic ERP analyses: a step-by-step tutorial review. *Brain topography* 20**,** 249-264.

Nichols, T.E., and Holmes, A.P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping* 15**,** 1-25.

Nunez-Elizalde, A.O., Huth, A.G., and Gallant, J.L. (2019). Voxelwise encoding models with non-spherical multivariate normal priors. *Neuroimage* 197**,** 482-492.

O'sullivan, J.A., Power, A.J., Mesgarani, N., Rajaram, S., Foxe, J.J., Shinn-Cunningham, B.G., Slaney, M., Shamma, S.A., and Lalor, E.C. (2015). Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cerebral Cortex* 25**,** 1697-1706.

O'sullivan, A.E., Crosse, M.J., Di Liberto, G.M., Cheveigné, A.D., and Lalor, E.C. (2021). Neurophysiological indices of audiovisual speech processing reveal a hierarchy of multisensory integration effects. *The Journal of Neuroscience***,** JN-RM-0906-0920.

O'sullivan, A.E., Crosse, M.J., Di Liberto, G.M., and Lalor, E.C. (2017). Visual Cortical Entrainment to Motion and Categorical Speech Features during Silent Lipreading. *Frontiers in human neuroscience* 10**,** 679.

Obleser, J., and Kayser, C. (2019). Neural entrainment and attentional selection in the listening brain. *Trends in cognitive sciences* 23**,** 913-926.

Poeppel, D. (2003). The analysis of speech in different temporal integration windows: cerebral lateralization as 'asymmetric sampling in time'. *Speech Communication* 41**,** 245-255.

Power, A.J., Mead, N., Barnes, L., and Goswami, U. (2013). Neural entrainment to rhythmic speech in children with developmental dyslexia. *Frontiers in human neuroscience* 7**,** 777.

Richards, B.A., Lillicrap, T.P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R.P., De Berker, A., and Ganguli, S. (2019). A deep learning framework for neuroscience. *Nature neuroscience* 22**,** 1761-1770.

Snyder, D., Chen, G., and Povey, D. (2015). Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Stevens, S.S. (1955). The measurement of loudness. *The Journal of the Acoustical Society of America* 27**,** 815-829.

Vellutino, F.R., Fletcher, J.M., Snowling, M.J., and Scanlon, D.M. (2004). Specific reading disability (dyslexia): What have we learned in the past four decades? *Journal of child psychology and psychiatry* 45**,** 2-40.

Winkler, I., Debener, S., Müller, K.-R., and Tangermann, M. (Year). "On the influence of high-pass filtering on ICA-based artifact reduction in EEG-ERP", in: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: IEEE), 4101-4105.

Wong, D.D., Fuglsang, S.A., Hjortkjær, J., Ceolini, E., Slaney, M., and De Cheveigne, A. (2018). A comparison of regularization methods in forward and backward models for auditory attention decoding. *Frontiers in neuroscience* 12**,** 531.

Wu, M.C.-K., David, S.V., and Gallant, J.L. (2006). Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.* 29**,** 477-505.

Yamins, D.L., and Dicarlo, J.J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience* 19**,** 356-365.

Zuk, N.J., Di Liberto, G.M., and Lalor, E.C. (2019). "Linear-nonlinear Bernoulli modeling for quantifying temporal coding of phonemes in brain responses to continuous speech", in: *2019 Conference on Cognitive Computational Neuroscience.* (Berlin, Germany).