Journal of Neural Engineering

OPEN ACCESS

RECEIVED
23 February 2022

REVISED
23 May 2022

ACCEPTED FOR PUBLICATION
16 June 2022

PUBLISHED
6 July 2022

PAPER

Robust decoding of the speech envelope from EEG recordings through deep neural networks

Mike Thornton[1], Danilo Mandic[2] and Tobias Reichenbach[3,*]

[1] Department of Computing, Imperial College London, London SW7 2RH, United Kingdom
[2] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2RH, United Kingdom
[3] Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-University Erlangen-Nürnberg, 91052 Erlangen, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: tobias.j.reichenbach@fau.de

Keywords: natural speech processing, EEG recordings, auditory attention decoding, deep neural networks

Abstract

Objective. Smart hearing aids which can decode the focus of a user's attention could considerably improve comprehension levels in noisy environments. Methods for decoding auditory attention from electroencapholography (EEG) have attracted considerable interest for this reason. Recent studies suggest that the integration of deep neural networks (DNNs) into existing auditory attention decoding (AAD) algorithms is highly beneficial, although it remains unclear whether these enhanced algorithms can perform robustly in different real-world scenarios. Therefore, we sought to characterise the performance of DNNs at reconstructing the envelope of an attended speech stream from EEG recordings in different listening conditions. In addition, given the relatively sparse availability of EEG data, we investigate possibility of applying subject-independent algorithms to EEG recorded from unseen individuals. *Approach.* Both linear models and nonlinear DNNs were employed to decode the envelope of clean speech from EEG recordings, with and without subject-specific information. The mean behaviour, as well as the variability of the reconstruction, was characterised for each model. We then trained subject-specific linear models and DNNs to reconstruct the envelope of speech in clean and noisy conditions, and investigated how well they performed in different listening scenarios. We also established that these models can be used to decode auditory attention in competing-speaker scenarios. *Main results.* The DNNs offered a considerable advantage over their linear analogue at reconstructing the envelope of clean speech. This advantage persisted even when subject-specific information was unavailable at the time of training. The same DNN architectures generalised to a distinct dataset, which contained EEG recorded under a variety of listening conditions. In competing-speakers and speech-in-noise conditions, the DNNs significantly outperformed the linear models. Finally, the DNNs offered a considerable improvement over the linear approach at decoding auditory attention in competing-speakers scenarios. *Significance.* We present the first detailed study into the extent to which DNNs can be employed for reconstructing the envelope of an attended speech stream. We conclusively demonstrate that DNNs improve the reconstruction of the attended speech envelope. The variance of the reconstruction error is shown to be similar for both DNNs and the linear model. DNNs therefore show promise for real-world AAD, since they perform well in multiple listening conditions and generalise to data recorded from unseen participants.

1. Introduction

Conventional hearing aids are known to provide only a limited benefit to their users, especially when operating in noisy conditions [1]. The ability to determine the focus of a user's attention could enable the development of smart hearing aids with improved outcomes for those who suffer with hearing loss. Recent studies have demonstrated that auditory attention in multi-speaker ('cocktail party') scenarios can be decoded noninvasively from electrophysiological recordings such as the electroencephalogram (EEG) [2–6]. One common paradigm for auditory attention decoding (AAD) is the method of backward linear modelling, whereby a set of coefficients are estimated in order to linearly reconstruct a speech feature from EEG recordings. In AAD applications, the speech feature is typically chosen to be the speech envelope, but other features can also be used, such as a waveform related to the fundamental frequency of speech [7, 8]. Both the speech envelope and the fundamental waveform of the attended speech stream are more strongly represented in a listener's EEG, and can be more accurately reconstructed from EEG recordings than corresponding features of the unattended speech streams. Therefore, in the backward modelling approach, a reconstruction score (typically Pearson's correlation coefficient between the reconstructed and the actual speech feature) for each speech stream serves as a marker of selective attention.

Since the processing in the auditory system is inherently nonlinear, it is natural to ask whether nonlinear methods for AAD can offer superior performance over linear methods. Nonlinear methods for backward modelling and AAD based on artificial neural networks have been introduced recently [9, 10]. Here, we set out to undertake a comprehensive account of the use of deep neural networks (DNNs) for backward nonlinear modelling of the speech envelope from EEG recordings. Artificial neural networks are heavily-parameterised, nonlinear models which are capable of representing a broad class of functions. In fact, they are universal function approximators [11]. DNNs are artificial neural networks which contain many layers of processing units (neurons). The correlation-based AAD technique described above can be also be used in conjunction with a DNN, by exchanging the linear backward model with a nonlinear DNN. Alternatively, auditory attention can be decoded directly without first reconstructing features from the EEG recordings, by utilising a DNN-based classifier which accepts EEG recordings as well as the candidate speech envelopes as inputs [10].

Nonlinear forward modelling based on DNNs has recently been employed to quantify the level of nonlinear processing that contributes to neural activity evoked by continuous speech [12]. The authors of that study found that as much as 25% of the evoked response arises due to nonlinear processes which can be captured by DNNs, thus justifying the use of DNNs for AAD. However, DNNs are known to suffer from issues surrounding generalisability. This has been highlighted by some recent investigations which did not achieve a competitive AAD performance across multiple datasets, when using DNNs which have elsewhere been reported to be effective [10, 13].

In this work, we compared the performance of two nonlinear DNNs as well as one linear model for predicting the speech envelope from EEG recordings. Following a recent study, we examined a fully-connected (FC) feed-forward neural network (FCNN) [9]. We also considered a more lightweight convolutional neural network (CNN) based on the EEGNet architecture, which has been proposed for a range of brain-computer-interface applications [14].

2. Materials and methods

2.1. Datasets and preprocessing

Two datasets from our research group were used in this work. The first dataset (termed Dataset 1 hereafter) was collected by Weissbart *et al* [15]. A total of 13 native English-speaking participants were instructed to attend to a single speaker narrating an audiobook in English, in noiseless and anechoic listening conditions. The EEG was recorded from all 13 participants, and each participant listened to 15 audiobook chapters in one recording session. The duration of each chapter was approximately 2.5 min, and each participant took breaks between chapters. This dataset therefore consists of 40 minutes of EEG responses to clean speech per participant. During the breaks, the participants were asked to answer a comprehension question in order to ensure attendance to the audiobook.

The second dataset (termed Dataset 2 hereafter) was collected by Etard and Reichenbach [16]. A total of 18 native English-speaking participants attended to a speaker narrating an audiobook chapter in several listening conditions: clean speech, speech in noisy conditions, and speech in competing-speaker scenarios. For the noisy speech, background babble noise was synthesised and combined with the speech at three different signal-to-noise ratios (SNRs) of 0.4 dB, −1.4 dB, and −3.2 dB. The comprehension levels for each SNR condition were 81%, 60%, and 34%, respectively, as measured through behavioural experiments. For the competing-speaker scenarios, two audiobooks were narrated simultaneously by a male and female speaker. There were two competing-speaker scenarios; in the first, the listener was instructed to attend to the male speaker whilst ignoring the female speaker, and in the second they were instructed to attend to the female speaker whilst ignoring the male speaker. Additionally, 12 of the participants listened to a speaker narrate an audiobook in a foreign language, Dutch. In this listening condition, the comprehension level was 0%. All stimuli were delivered binaurally. As with Dataset 1, the participants were asked comprehension questions in order to ensure attendance to the target audiobook. For each listening condition, the EEG was recorded in four trials of approximately 2.5 min in duration. This dataset therefore consists of 10 min of EEG recorded for each listening condition per participant.

In both datasets, 64-channel scalp EEG was recorded with the same equipment inside the same anechoic chamber. The EEG was sampled at a rate of 1 kHz, with electrodes positioned according to the standard 10–20 system via the actiCAP electrode cap (BrainProducts, Germany). The EEG signals were then amplified and digitised with the actiCHamp amplifier (BrainProducts, Germany). In Dataset 1, the left earlobe was used as the physical EEG reference, whereas the right earlobe was used for this purpose in Dataset 2. In order to align the stimulus with the recorded EEG, the acoustic adapter for StimTrack (BrainProducts, Germany) was used to record the audio at 1 kHz whilst simultaneously presenting it to the participant (at 44.1 kHz). The resulting sound channel was used to align the original audio tracks with the EEG recordings during post-processing.

Preprocessing was performed using default routines available in MNE-Python version 0.24.1 [17]. To obtain the speech envelopes, we computed the absolute value of the Hilbert transform of each speech stream. The speech envelopes were low-pass filtered below 50 Hz (linear phase type 1 FIR anti-aliasing filter, Hamming window, 12.5 Hz transition bandwidth, −6 dB attenuation at 56.25 Hz, −53 dB stopband attenuation) and resampled to 125 Hz. To preprocess the EEG recordings, all channels were low-pass filtered below one of several upper passband edges (linear phase type 1 FIR anti-aliasing filters, Hamming windows, −53 dB stopband attenuation). The considered upper passband edges were: 8 Hz (order 1651, 2 Hz transition bandwidth, −6 dB attenuation at 9 Hz), 12 Hz (order 1101, 3 Hz transition bandwidth, −6 dB attenuation at 13.5 Hz), 16 Hz (order 825, 4 Hz transition bandwidth, −6 dB attenuation at 18 Hz) and 32 Hz (order 413, 8 Hz transition bandwidth, −6 dB attenuation at 36 Hz). The EEG recordings were subsequently resampled to 125 Hz and high-pass filtered above one of two lower passband edges in order to remove slow drifts (linear phase type 1 FIR filters, Hamming windows, −53 dB stopband attenuation): 0.5 Hz (order 825, 0.5 Hz transition bandwidth, −6 dB attenuation at 0.25 Hz), or 2 Hz (order 207, 1 Hz transition bandwidth, −6 dB attenuation at 1.5 Hz). Finally, for every trial, each EEG channel was standardised to have zero mean and unit variance.

2.2. Linear models

A linear backward model can be specified in the time domain by a matrix of parameters $\theta_{i,j}$. These are convolved with the EEG recordings to produce an estimate of the speech envelope:

$$\hat{y}_t = \sum_{i=1}^{C} \sum_{j=0}^{L-1} x_{t-j,i}\theta_{i,j}. \tag{1}$$

In this expression, $\hat{y}_t$ denotes an estimate of the speech envelope sampled at time t, $x_{t,i}$ designates the EEG sampled at time t from electrode i, C represents the number of EEG channels being considered, and L is the filter length which describes how many temporal EEG samples are employed to estimate the speech envelope. In other words, the speech envelope is represented as a linear combination of the EEG recordings $x_{t-j,i}$, which are weighted by the parameters $\theta_{i,j}$.

The parameters of the linear model are obtained by minimising the sum of squared errors $\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$, where T is the total number of time samples available in the training dataset. We used ridge regression, which employs an L2 regularisation term $\lambda \sum_{i=1}^{C} \sum_{j=0} \theta_{i,j}^2$ within the objective function. This results in a better-posed regression problem which is less susceptible to overfitting [18, 19]. The L2 penalty penalises large weights, and the strength of the penalty is controlled by the hyperparameter λ.

2.3. DNNs

The linear models in (1) depend on $L \times C$ parameters. Nonlinear models implemented as DNNs typically employ a much larger set of parameters and possess a more complicated functional form. The DNNs in this work can therefore be considered as more general functions relating the EEG recordings and the speech envelope.

The fundamental unit of any neural network is the ‘neuron’. A neuron receives a pre-determined number of inputs, which it linearly combines according to its set of parameters (or weights). A nonlinear activation function is then applied to the resulting scalar quantity. Common choices for the activation function are the sigmoid and the hyperbolic tangent, as well as the rectified linear unit (ReLU) [20]. The latter is defined as the function $f(x) = x$ if $x > 0$ and $f(x) = 0$ otherwise.

A feed-forward neural network consists of layers of neurons, with neurons in a particular layer receiving as inputs the outputs of neurons in preceding layers (neurons within a single layer do not connect with one another). If each neuron in a particular layer is connected with each neuron in the preceding layer, the neural network is described as ‘fully connected’ (FC). The fully-connected feed-forward neural network (FCNN) used in this work is depicted in figure 1. If each neuron in a particular layer is instead only connected to a neighbourhood of input neurons, and all neurons in the same layer share the same parameters (weight sharing), then the neural network is described as a convolutional neural network (CNN). The CNN that was used in this work is shown in figure 1. Other types of connectivities exist, including skip connections (e.g. residual neural networks [21]) and feedback connections (recurrent neural networks [11]).

The FCNN used in this work was inspired by the architecture used by de Taillez *et al* [9]. A spatiotemporal segment of EEG recordings is passed through several FC feed-forward layers, with each layer containing fewer neurons than the preceding layer. The activation function is the hyperbolic tangent. The

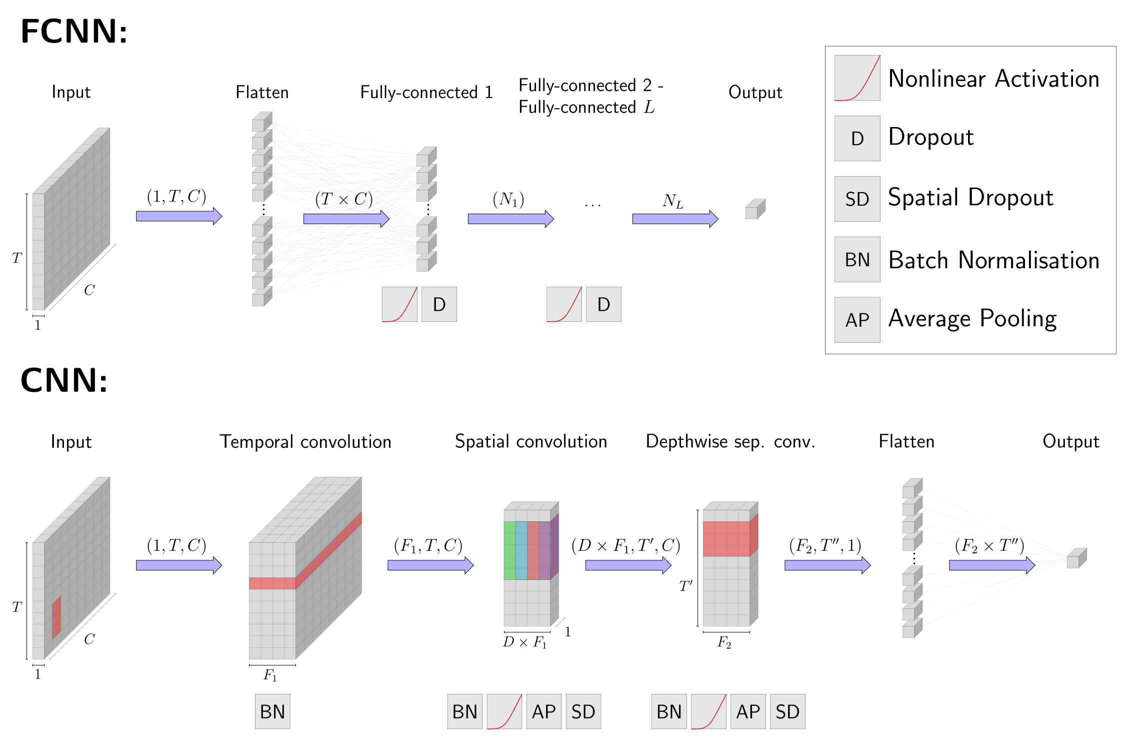


Figure 1. The two neural network architectures used in this work. A spatiotemporal segment of T EEG samples and C channels is presented to both of the neural networks. Top: the FCNN architecture consists of several fully-connected hidden layers. The Lth hidden layer consists of N_{D_L} neurons. A nonlinear activation function is applied to the output of each hidden layer, followed by a dropout layer. Bottom: the CNN architecture employs convolutional layers, which make use of local connectivity and weight-sharing to reduce the number of parameters in the network. The 1st convolutional layer consists of F_1 convolutional filters, each sharing a common input but comprising distinct parameterisations. The 2nd convolution flattens the channel dimension, and consists of $D \times F_1$ convolutional filters. The 3rd convolutional layer implements the so-called depthwise separable convolution, which is similar to an ordinary convolutional layer consisting of F_2 low-rank convolutional filters. Following [14], we set $F_2 = D \times F_1$ to reduce the dimensionality of the hyperparameter search. Several operations may be applied to the activations of each convolutional layer, including a non-linear activation function, batch normalisation, and spatial dropout. Average pooling is a form of downsampling whereby the activations of a neighbourhood of neurons are replaced by the average activation. This form of downsampling is only applied along the temporal dimension.

number of inputs is equal to $C \times T$, with C as the number of EEG channels used, and T as the number of temporal samples in the segment. The scalar output represents a point estimate of the speech envelope at the onset of the segment. Following de Taillez *et al*, the number of neurons in each hidden layer decreases linearly from $C \times T$ to 1. The number of hidden layers is a tunable parameter.

In this work, two types of regularisation are used to help control overfitting. The first type, dropout, randomly sets the activation of some neurons to zero according to some probability [22]. This regulates how strongly the neural network can depend on the activation of any particular neuron. The second type, L2 regularisation or 'weight decay', uses a term $\lambda \sum_{k=1}^{N} |w_k|^2$ within the objective function, where w_k denotes the kth parameter of the neural network. This term penalises neural networks for which some weights are much larger than others, and promotes neural networks which do not rely too much on any particular neuron. The tunable hyperparameter λ controls the strength of the regularisation penalty.

The number of weights required to fully connect two adjacent layers consisting respectively of N_1 and N_2 neurons is $N_1 \times N_2$ or $N_1 \times N_2 + N_2$, depending on whether a bias term is used. The number of parameters in an FCNN therefore grows quickly with the number hidden layers, and the FCNN can become over-parameterised. Due to local connectivity, CNNs can often represent similar functions to FCNNs with far fewer parameters, thus helping to prevent overfitting. We therefore investigated the performance of a CNN at reconstructing the speech envelope from EEG recordings.

Our choice of CNN was inspired by the EEGNet architecture of Lawhern *et al* [14], which employs the exponential linear unit (ELU) as a nonlinear activation function, as well as batch normalisation and average pooling [23]. Batch normalisation improves convergence during training by making the optimisation problem smoother [24, 25]. Average pooling is a form of downsampling, in which the average activation of a neighbourhood of neurons is taken and used as the input to the next layer. To regularise the CNN, we used L2 regularisation and a variant of dropout known as spatial dropout [26], whereby entire weight-sharing layers are dropped from the training process according to some probability. This technique can be more effective than ordinary dropout for training CNNs, since weight sharing dilutes the effect of dropping the activation of individual neurons. The CNN architecture includes a ‘depthwise separable convolution’ layer, which utilises low-rank approximations to ordinary convolutional filters. The scalar output of the CNN is formed by taking a linear combination of all of the activations in the final convolutional layer.

2.4. Training procedure

For both of the DNNs as well as the linear model, the spatio-temporal input segment consisted of 63 EEG channels and 50 time samples (C=63; T=50). The temporal length of the segment was therefore 400 ms, since a sampling rate of 125 Hz was used. For the CNN, average pooling was performed across the temporal axis using a neighbourhood of two neurons after the spatial convolution layer, and a neighbourhood of five neurons after the depthwise convolution layer. Therefore, the values of the temporal dimensions T′ and T″ in figure 1 were 25 and 5, respectively.

The coefficients of the linear model were fitted through ridge regression, as discussed in section 2.2. Ridge regression permits a simple closed-form expression for the optimal (least-squared-error) coefficients, given a training dataset and a regularisation parameter. In contrast, DNNs can rarely be solved analytically, and gradient-descent methods are commonly used to train them (that is, to fit their parameters). Following [9], in this work we optimised the DNN parameters by minimising the negative correlation coefficient between the reconstructed speech envelope and the target speech envelope. The NAdam optimiser was used, which employs adaptive step sizing and accelerated gradient descent through a Nesterov-like momentum term [27].

Dataset 1 consisted of 15 trials per participant, each of approximately 2.5 min in duration. We reserved nine of these trials for model training, three for validation, and three for evaluation. Dataset 2 consisted for four trials per listening condition, per participant. Each trial had a duration of approximately 2.5 min. We used eight trials for model training (four clean-speech trials and four high-SNR speech-in-noise trials), and four trials for validation (from the low-SNR speech-in-noise condition). The remaining trials were used for evaluation.

During training, batches of EEG data were presented to the DNNs, and a corresponding batch of predicted speech envelope values was produced. These were correlated against the actual speech envelope values, and the DNN parameters were updated via a NAdam gradient descent step in order to maximise the correlation coefficient. After iterating through all batches of data (one epoch), the correlation score was evaluated on the validation dataset. An early-stopping procedure was used with a patience factor of P: if the validation correlation score did not increase within P consecutive epochs, training was terminated. Otherwise, the process was repeated for another epoch. The model parameters which produced the highest validation correlation score were saved. During hyperparameter tuning, P was set to 3. Once the hyperparameters were fixed, the DNNs were trained with an increased patience factor of 5.

For each analysis, we trained 15 linear models with different regularisation parameters spaced evenly on a logarithmic scale (ranging from 10^{-7} to 10^{7} inclusive). The model that achieved the highest correlation score on the validation dataset was selected for testing.

The DNN hyperparameters were tuned by randomly sampling 80 hyperparameter configurations (random search), and the configuration that led to the highest validation score (correlation coefficient) was selected for testing. The DNN hyperparameters included an L2 regularisation (weight decay) parameter, the initial optimiser step size (learning rate), the number of hidden layers or convolutional filters, and the number of filters belonging to each convolutional layer. We only tuned these hyperparameters once per DNN, for the population models trained using Dataset 1.

For the FCNN, the parameters of the random search were as follows: batch size = (64, 128, or 256); weight decay = (10^{-8}, 10^{-7}, ..., or 10^{-2}); number of hidden layers = (1, 2, 3 or 4); weight decay = (10^{-8}, 10^{-7}, ..., or 10^{-2}); initial learning rate = (10^{-6}, 10^{-5}, ..., or 10^{-2}). The dropout rate was a real number sampled uniformly between 0 and 0.5.

For the CNN, the parameters of the random search are as follows: batch size = (64, 128, or 256); weight decay = (10^{-8}, 10^{-7}, ..., or 10^{-2}); initial learning rate = (10^{-6}, 10^{-5}, ..., or 10^{-2}). The spatial dropout rate was a real number sampled uniformly between 0 and 0.4. In order to reduce the dimensionality of the random search we used the condition $F_2 = F_1 \times D$, with D = (2, 4, or 8) and F_1 = (2, 4, or 8).

Since the task of fitting subject-specific models is quite different to fitting a population model, the optimal hyperparameters for each subject-specific DNN might vary. In this work, we re-tuned the initial learning for each subject-specific DNN using a similar random search procedure, whilst holding the other hyperparameters fixed. For the FCNN population model, we found that the following hyperparameters were suitable: three hidden layers; a dropout rate of 0.45; a batch size of 256; and a weight decay value of 1×10^{-4}. Therefore, the number of neurons in each hidden layer were, in order, 2363, 1576, and 788. For the CNN population model, we found the following parameters to be suitable: $F1 = 8$; $D = 8$; a spatial dropout rate of 0.20; a batch size of 256; and a weight decay value of 1×10^{-8}. Recall from figure 1 that $F1$ is the number of convolutional filters in the first convolutional layer, and $F1 \times D$ is the number of convolutional filters in each of the second and third convolutional layers.

2.5. Analysis procedure

To evaluate the models, the EEG data were split into contiguous windows without overlap. Window sizes of 250 samples (two seconds) were considered unless otherwise stated. As in the training step, the predicted speech envelope values in each window were correlated against the actual speech envelope values. The mean and variance of the correlation score over all windows were then calculated, since these quantities are of interest in AAD applications. To construct a null distribution, the predictions for each window were also correlated against the true speech envelope in unrelated windows.

In section 3.4 we applied the linear model as well as both DNNs to EEG recorded in competing-speakers scenarios. The correlation-based method was used to decode auditory attention using each DNN or the linear model. The performance of each backward model was quantified through the attention decoding accuracy for a given window size W. The information transfer rate (bit rate) B was also calculated according to [28, 29]:

$$WB = \log_2 N + P\log_2 P + (1 - P)\log_2 \frac{1 - P}{N - 1}, \quad (2)$$

where N is the number classes in the classification problem (two in this case), and P is the attention decoding accuracy. The bitrate in (2) is scaled by the latency of the decoder, W, to obtain an effective bitrate that takes into account the temporal resolution of the decoder. In this work, W was calculated by adding the duration of the temporal receptive field (T, which corresponds to 0.4 s in this study) to the duration of the window which was used to calculate the correlation coefficients.

The neural networks were implemented in PyTorch version 1.10.0 [30]. Statistical analyses were conducted using Scipy version 1.7.1 and Statsmodels version 0.11.1 [31, 32].

3. Results

3.1. Subject-specific models

In Dataset 1, thirteen participants listened to a single speaker who narrated an audiobook in English without background noise. The participants' EEG was recorded from 63 scalp channels whilst they listened. For our first analysis, we fitted linear and nonlinear models to each participant's EEG in order to reconstruct the envelope of the speech stream. We tested the performance of the models by dividing the test data into windows of a duration of two seconds. The reconstructed speech stream was subsequently correlated against the actual speech stream in each window. We performed this procedure using several different EEG frequency bands, and we found that using the 0.5–8 Hz band yielded linear decoders with the greatest reconstruction scores (Pearson correlation coefficients) (figure 2(a)). For this frequency band, the spread of reconstruction scores for all participants is reported in figure 2(b). We used this frequency band for all subsequent analyses.

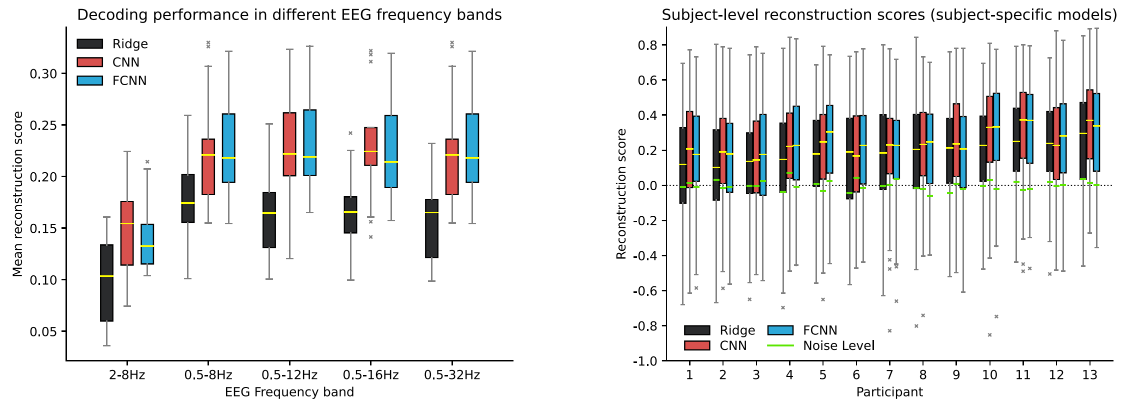
Null distributions for the reconstruction scores were obtained by correlating each reconstructed speech envelope with the true speech envelope from an unrelated window. The median values of the null distributions are shown in green on figure 2(b). Since the reconstruction scores and null distributions were approximately normally distributed, we tested the reconstruction scores for significance with a t-test (single-tailed unpaired t-test, FDR-corrected). In addition, we compared the reconstruction scores of each pair of models for every participant (paired t-tests, FDR corrected). The corrected p-values are reported in table 1. All of the models achieved significant reconstruction scores for every participant. There were somewhat significant differences between the performances of the two DNNs for 5 participants. The CNN (FCNN) outperformed the linear model with significance for 11 (9) of the participants.

To analyze the performance on the population level, we calculated the mean reconstruction score for every participant and model. We then compared the 13 mean reconstruction scores achieved by each model. We found no significant difference between the two DNNs ($p = 0.91$, two-tailed paired t-test). However, both DNNs significantly outperformed the linear model (CNN: $p = 1.1 \times 10^{-04}$; FCNN: $p = 2.1 \times 10^{-04}$; single-tailed paired t-tests, Bonferroni corrected.)

The mean and standard deviation of the reconstruction score varied with window duration. We determined the dependence of the mean and standard deviation of the reconstruction scores on the window duration by performing the analysis procedure with windows of various sizes (ranging from 0.1 s and 10 s). The mean and standard deviation of the reconstruction scores were averaged over all participants (figure 3). The mean reconstruction scores of both linear and nonlinear models are strongly degraded for window sizes less than 2 s. For window sizes greater than 2 s, the mean reconstruction score for each DNN was around 30% above that of the linear model. The mean of the set of 13 standard deviations was very similar for all three methods across all window sizes.

3.2. Subject-independent models

To test whether the models generalise between participants, we left one participant's data out of the training procedure, and instead trained each of the models on the data from the 12 remaining participants. We then repeated this process 13 times, leaving out a different participant each time. In this way, we trained 13 subject-independent models and applied them to data from the unseen participant. For comparison, we also trained population models using training data from all of the participants, and applied these to distinct test data (recorded from the same 13 participants). Our results are summarised in figure 4.



(a) (b)

Figure 2. Two distinct DNN architectures, as well as a linear model, were used to relate EEG recordings to the envelope of clean speech. The decoding performance for all three methods when different EEG frequency bands are used is shown in (a). Each boxplot comprises the mean envelope reconstruction score (correlation coefficient) for each participant. The subject-level results for the EEG frequency band 0.5–8 Hz are depicted in (b). Each boxplot represents the median and range of the reconstruction score when a 2 s correlation window is employed. The median reconstruction scores of the null distributions are shown in green.

On the subject level, the use of the linear subject-independent models resulted in significant mean reconstruction scores for 9 of the 13 participants (single-tailed unpaired t-test, FDR corrected). Both the subject-independent CNN and FCNN yielded significant reconstruction scores for 12 participants. For each participant, we compared the use of each pair of subject-independent models using paired t-tests (FDR corrected). The CNN and FCNN did not perform significantly differently for any of the 13 participants. The CNN (FCNN) outperformed the linear method with significance for 1 (3) participants. On the population level, both subject-independent DNNs significantly outperformed the subject-independent linear models (CNN: $p = 9.2 \times 10^{-5}$; FCNN: $p = 0.01$; single-tailed t-tests, Bonferroni corrected). There was no significant difference between the subject-independent DNNs on the population level.

The subject-independent decoders yielded scores which were approximately 50% below those of the subject-specific decoders. The population decoders performed better than the subject-independent decoders, but worse than the subject-specific decoders. This is to be expected, since the subject-specific and subject-independent decoders respectively represent the two extremes in which either there is only subject-specific information available, or there is no subject-specific information available.

3.3. Performance of subject-specific models in different listening conditions

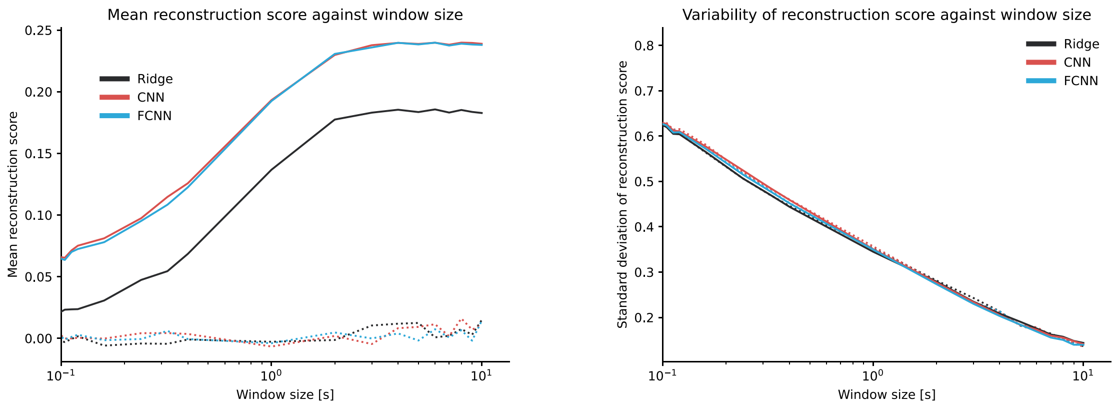
For real-world applications, a decoder needs to perform well across a range of listening conditions. We therefore trained subject-specific decoders using Dataset 2, which consisted of EEG recorded under a number of different listening conditions. We used the clean speech in native English, as well as speech in the highest SNR condition (0.4 dB) to train the decoders. We used the lowest SNR condition (−3.2 dB) to validate the training procedure, and we evaluated the trained decoders on the medium SNR condition (−1.4 dB), as well as on competing-speaker scenarios, and the clean speech in foreign Dutch condition (for which the comprehension level was 0%). For each method, we calculated the mean reconstruction score for each participant. The spread of mean reconstruction scores in each condition are shown in figure 5. To compare the performance of the trained models in each listening condition, we compared the sets of mean reconstruction scores achieved by each pair of models within each listening condition using two-tailed paired t-tests (FDR corrected). There was no significant difference between the DNNs in any listening condition. However, both DNNs significantly outperformed the linear model at reconstructing the attended speech stream in the competing-speakers conditions, as well as in the background babble noise condition. The DNNs performed similarly to the linear models at reconstructing the envelope of clean speech in a foreign language, as well as at reconstructing the unattended speech envelope in the competing-speakers conditions.

3.4. Attention decoding performance

For our final case study, we investigated whether the subject-specific decoders described in section 3.3 could actually be used for AAD in the competing-speaker scenarios. We compared the reconstruction score (correlation coefficient) for the attended and unattended speakers in each window, and counted how many times the reconstruction of the attended envelope was greater than that of the unattended envelope. This number was taken to be the number of correct classifications, from which the binary classification accuracy could be directly calculated

Table 1. Statistical tests on the data shown in figure 2(b). The first three rows show the *p*-values obtained when testing for difference between the reconstruction scores (ρ) and null distributions (ρ_0) for each method. These were FDR-corrected independently of the *p*-values in the next three rows. The next three rows show the *p*-values obtained by testing for differences between the reconstruction scores of the different techniques. P-values smaller than 0.05 are presented in boldface.

Alternative	Participant												
hypothesis	1	2	3	4	5	6	7	8	9	10	11	12	13
$\rho^{\text{ridge}} > \rho_0^{\text{ridge}}$	**2.94×10^{-05}**	**2.77×10^{-04}**	**1.02×10^{-03}**	**1.48×10^{-09}**	**7.86×10^{-13}**	**1.23×10^{-06}**	**1.89×10^{-08}**	**1.61×10^{-09}**	**2.05×10^{-13}**	**1.10×10^{-13}**	**2.01×10^{-19}**	**2.57×10^{-19}**	**3.79×10^{-19}**
$\rho^{\text{CNN}} > \rho_0^{\text{CNN}}$	**2.12×10^{-11}**	**1.75×10^{-10}**	**1.36×10^{-08}**	**5.32×10^{-19}**	**1.80×10^{-12}**	**5.01×10^{-10}**	**5.43×10^{-16}**	**1.16×10^{-18}**	**9.80×10^{-17}**	**4.62×10^{-24}**	**1.24×10^{-28}**	**2.32×10^{-18}**	**1.13×10^{-32}**
$\rho^{\text{FCNN}} > \rho_0^{\text{FCNN}}$	**1.90×10^{-14}**	**1.37×10^{-11}**	**1.98×10^{-09}**	**3.71×10^{-16}**	**1.56×10^{-18}**	**2.02×10^{-11}**	**7.56×10^{-18}**	**2.95×10^{-16}**	**2.99×10^{-12}**	**1.24×10^{-28}**	**1.32×10^{-26}**	**3.58×10^{-23}**	**2.53×10^{-21}**
$\rho^{\text{CNN}} \neq \rho^{\text{FCNN}}$	3.75×10^{-01}	**4.41×10^{-02}**	7.05×10^{-01}	3.75×10^{-01}	**1.31×10^{-04}**	**3.11×10^{-02}**	2.90×10^{-01}	6.29×10^{-01}	**3.95×10^{-04}**	5.68×10^{-01}	4.44×10^{-01}	3.75×10^{-01}	**1.25×10^{-02}**
$\rho^{\text{CNN}} > \rho^{\text{ridge}}$	**1.18×10^{-04}**	**1.18×10^{-04}**	**4.93×10^{-02}**	**9.54×10^{-04}**	**2.80×10^{-02}**	4.18×10^{-01}	**4.93×10^{-02}**	**4.93×10^{-02}**	**7.69×10^{-03}**	**2.66×10^{-08}**	**3.27×10^{-07}**	7.11×10^{-01}	**1.09×10^{-04}**
$\rho^{\text{FCNN}} > \rho^{\text{ridge}}$	**1.00×10^{-06}**	**1.25×10^{-02}**	**3.11×10^{-02}**	**6.97×10^{-05}**	**1.11×10^{-08}**	**2.59×10^{-02}**	1.25×10^{-01}	9.01×10^{-02}	5.85×10^{-01}	**1.11×10^{-08}**	**3.44×10^{-07}**	4.18×10^{-01}	**7.06×10^{-03}**



(a) (b)

Figure 3. The mean (a) and standard deviation (b) of the reconstruction score (Pearson's correlation coefficient), averaged over all participants from Dataset 1, is plotted against the size of the correlation window used during evaluation. For window sizes less than 2 s in duration, the mean reconstruction score was considerably degraded. The variability of the reconstruction score increases sharply with decreasing window size, but is similar for all three methods. The dotted lines show the mean and standard deviation of the null reconstruction scores for the three methods.

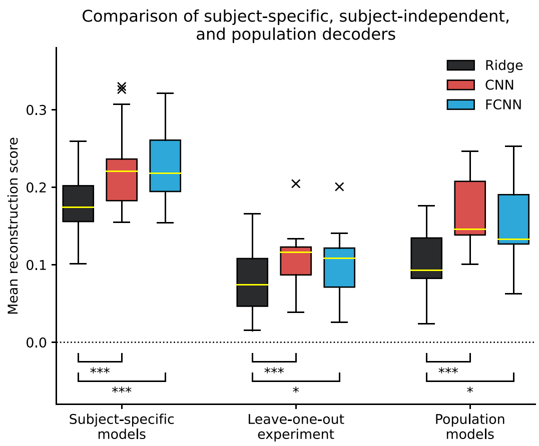


Figure 4. Comparison between subject-specific and subject-independent decoders applied to clean speech (Dataset 1). The boxplots represent the spread of the mean reconstruction scores achieved for each participant. The first group (subject-specific models) shows the mean reconstruction scores achieved by subject-specific decoders applied to Dataset 1. The second group (leave-one-subject-out experiment) shows the mean reconstruction scores achieved by the subject-independent decoders described in section 3.2. For the third group (population models), population models were trained with Dataset 1 and subsequently applied them to data recorded from individual subjects in Dataset 2. Statistical significance is denoted by asterisks (none, p >=0.5; *, 0.5 > p >= 0.1; **, 0.1 > p >=0.01; *** , 0.01 > p >= 0.001; ****, 0.001 > p >=0.0001; *****, p < 0.0001).

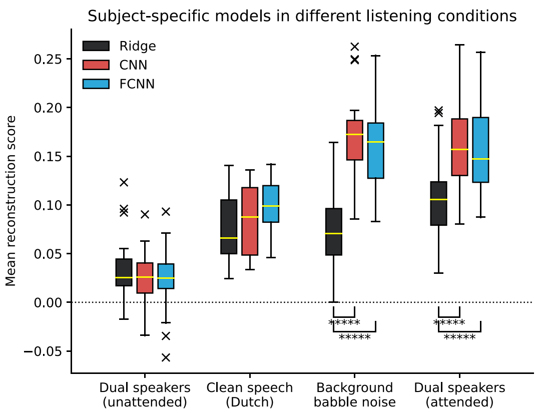


Figure 5. The spread of mean reconstruction scores for the participants in Dataset 2 when subject-specific decoders were applied in different listening conditions. Statistical significance is denoted through asterisks as detailed in the caption of figure 4.

as a percentage of the total number of windows in the trial. The decoding accuracies for three different window durations (2 s, 5 s, 10 s) are shown in figure 6. Both DNNs offered clear accuracy improvements over the linear model across all three window durations.

Following [9], we also calculated effective bitrates for attention decoding using different window sizes. The bitrate is related to the time-rate of correct classifications, and was calculated according to equation (2). We found that a window size of 2 s maximises the decoding performance of the CNN as well as that of the linear model (figure 6(b)). A window size of 5 s was marginally more suitable for the FCNN. Both DNNs achieved much higher bitrates than the linear model.

4. Discussion

We have investigated the performance of two types of DNNs at estimating the speech envelope from EEG recordings. The performance of each DNN was compared to that of a standard linear model. A comprehensive evaluation has shown that the two DNNs can achieve very similar performances when reconstructing the envelope of clean speech, whilst exceeding that of the linear model by about 30%. The advantage

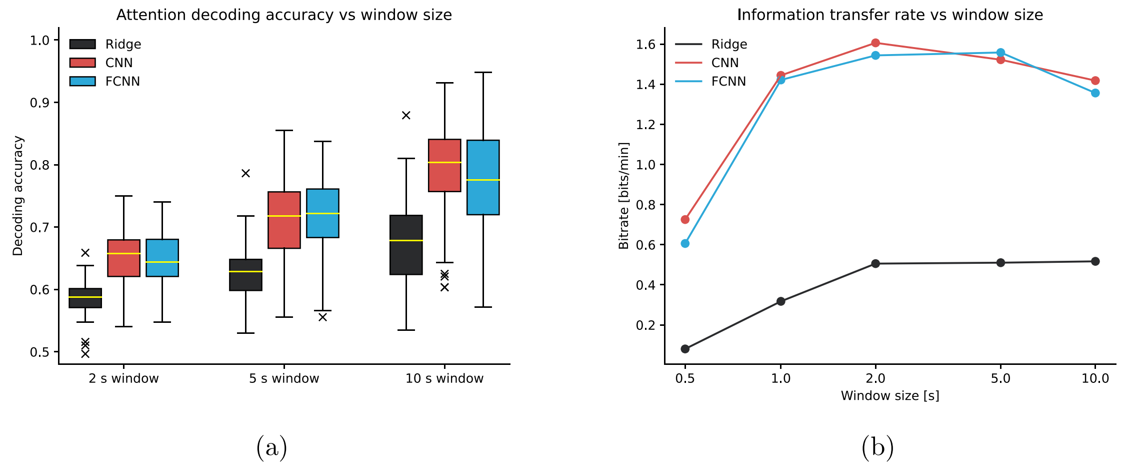


Figure 6. Comparison of the attention decoding accuracies of the three methods, across three different window sizes (a). The information transfer rate for the three methods, which quantifies the tradeoff between decoding accuracy and window duration (b).

of using the DNNs over the linear models persisted even when the models were applied to subjects whose EEG data had not been seen during training. Importantly, the DNN architectures and hyperparameters have been shown to generalise to a distinct data set, and subject-specific models have been applied effectively to data in which speech was presented in different types of noise. Our results have demonstrated that DNNs have the ability to robustly enhance the decoding of speech features from EEG recordings.

4.1. Deep learning methodology

To use the DNNs effectively, some special steps were taken. Firstly, rather than presenting batches of consecutive EEG windows to the DNNs during training, we shuffled the order of the windows in the training dataset. This is a departure from the approach suggested in de Taillez *et al* [9]. We also used much smaller batch sizes to train the DNNs. Small batch sizes are often desirable in deep learning, since they can help to avoid overfitting via noise injection [33, 34]. We note that a much larger batch size of 1024 samples was used in the study by Cicarelli *et al* [10], and no weight decay was employed. In comparison to other studies, we performed a more complete hyperparameter search, which included a search over L2 regularisation parameters, training batch sizes, initial learning rates, dropout rates, and the number of hidden layers/ convolutional filters. Furthermore, we re-tuned the initial learning rate for each subject-specific DNN and participant, whilst keeping the other hyperparameters fixed. Future work will explore further individualisation of additional hyperparameters, for example the weight decay parameter. In our study, we found that it was most effective to use three hidden layers within the FCNN, whereas previous studies have made use of just one hidden layer [9, 10].

The power of EEG signals vary between participants, and may vary over time and between recording sessions for a single participant. This is of little consequence for trained linear models, the outputs of which are equivariant with respect to scalings of the inputs (thus leaving correlation-based reconstruction scores invariant). However, the scores achieved by the DNNs are sensitive to changes in the power of the input channels, due to the nonlinear activation functions employed by the DNNs. In order to fix the power of the inputs, we standardised each EEG channel to have zero mean and unit variance, for all subjects in all trials. This re-scaling is non-causal, and for real-world applications it may be preferable to standardise the EEG recordings by first removing the mean via high-pass filtering, and then normalising the power of each EEG channel by dividing the recordings by a rolling estimate of the standard deviation. Full-cap EEG signals may alternatively be standardised across the channel axis, i.e. by removing the mean and dividing by the standard deviation of all EEG signals at each sample. However, this method would not be appropriate for low-density wearable montages such as concealed EEG or ear-EEG [5, 6]. We note that the calculation of the speech envelope via the Hilbert transform is also a non-causal operation which must be adapted for real-time applications.

The scores achieved by the two DNNs investigated in this work were remarkably close. In fact, the outputs of the two neural networks were themselves highly correlated, which suggests that the two DNNs had learned to represent very similar functions. Owing to local neuron connectivity, the CNN required far fewer parameters than the FCNN: about nine thousand versus twelve million, respectively. The linear model required about three thousand parameters, which is comparable to the number of parameters in the CNN. The CNN may therefore be a preferable, lightweight alternative to the FCNN for practical applications. Future investigation may reveal effective architectures which are even more lightweight, for example by removing or 'pruning' neurons which are of low importance [35, 36]. Additionally, prior information about this signal processing problem could be exploited by imposing inductive biases on a DNN [37]. For example, the neural response to the speech envelope has been well characterised in the literature, and the spatial arrangement of the EEG sensors is known by the experimentalist.

4.2. Subject-specific decoders

The existing literature surrounding DNNs for AAD focuses almost exclusively on the attention decoding accuracy of the correlation-based algorithm, with DNNs being used to reconstruct the attended speech stream. It is natural to also investigate how well the DNNs perform at the fundamental task of reconstructing the attended speech stream. In order to do this, we began by training subject-specific models to predict the envelope of clean speech from EEG recordings. The effect of broadening the EEG frequency band from 0.5–8 Hz to 0.5–32 Hz had no discernible impact on the performance of the DNNs, as measured via Pearson's correlation coefficient between the actual and reconstructed speech envelopes. However, the use of the frequency band 0.5–8 Hz in place of the frequency band 2–8 Hz led to considerably improved results when decoding the envelope of clean speech. De Taillez et al found it beneficial to use broadband EEG signals in the range 1–32 Hz instead of signals in the range 2–8 Hz, since this resulted in a higher information transfer rate when using DNNs to decode auditory attention [9]. It is likely that much of this benefit can be attributed to the incorporation of lower-frequency components of the EEG signals. The effect of incorporating higher-frequency EEG components on the attention decoding accuracy in competing-speakers conditions cannot be directly inferred from our analysis, since we only considered the effect of the spectral content of the EEG signals in the context of reconstructing the envelope of clean speech in quiet conditions.

We used the 0.5–8 Hz EEG frequency band for subsequent analyses. Both of the DNNs as well as the linear modelling method achieved significant reconstruction scores for all participants. On the population level, the improvement offered by the DNNs was statistically significant. The overall performance of the DNNs was around 30% greater than that of the linear model. Even on the subject level, the CNN (FCNN) offered a statistically significant performance increase compared against the linear model for 11 (9) participants.

We found that for windows smaller than around 2 s in duration, the reconstruction accuracy of all three methods was severely degraded. The latency of a real-world decoder which is based on the correlation method may therefore be limited to this timescale, unless techniques such as state-space models are employed [3]. Indeed, we found that a window size of about 2 s maximises the information transfer rate of the correlation-based AAD algorithm. The variability in the reconstruction score followed similar power-law dependencies on window size for all three methods. This finding contrasts with a previous study which found that the reconstruction score of a DNN similar to the FCNN used in this work was much more variable than that of a linear model, when applied in a competing-speaker scenario [38].

4.3. Subject-independent decoders

Using Dataset 1, we trained 13 versions of each DNN and linear model to reconstruct the envelope of clean speech from EEG recordings. Each version was obtained by leaving out one of the thirteen participants during training. The DNNs and the linear model were then applied to the data of the 'unseen' participant. This allowed us to compare the performance of subject-independent methods with subject-specific methods whilst holding constant certain factors such as the experimentalist, the experimental protocol, the stimuli, and the duration of the experiment.

All three subject-independent decoders yielded reconstruction scores that were significantly different from the null distribution for the majority of the participants (9 for the linear model; 12 for both the CNN and the FCNN). We found that all three subject-independent decoders performed very similarly on the subject level. However, on the population level, the subject-independent DNNs both significantly outperformed the subject-independent linear model.

The subject-independent decoders performed significantly worse than their subject-specific counterparts (the performance decrease was around 50% for all three methods). Whilst a performance penalty is to be expected when subject-independent information is unavailable, a penalty of this magnitude may imply that some subject-specific information is required for real-world applications.

4.4. Application to EEG recorded under different listening conditions

For real-world decoding applications, the decoder must perform well across a variety of listening conditions. It was recently found that linear models can assess neural speech tracking in two-speaker scenarios in a manner that is robust against distortions to the two speech streams [39]. Our investigation furthers this research by studying how well linear and nonlinear models can assess neural speech tracking in clean and noisy listening conditions with varying levels of speech clarity and comprehension.

Decoders for auditory attention to one of two competing speakers are usually trained on the EEG data obtained when the participants listen to two competing talkers [9, 10]. Here, due to the limited amount of competing-speakers data available in Dataset 2, we used a somewhat different approach in which we trained linear and nonlinear subject-specific decoders using a mixture of clean speech and speech-in-babble-noise conditions. The single-speaker conditions used to train the decoder provide a more stable teaching signal than the competing-speaker conditions, in which participants may sometimes direct their attention to the speaker labelled `unattended'. However, the clean- and speech-in-noise- single-speaker conditions may not elicit the same attention dynamics that are exhibited in the competing-speakers conditions. It was therefore important to find that our DNNs were able to reconstruct the attended speech envelope in the competing-speakers conditions as well. It is likely that even greater reconstruction scores could be achieved if competing-speakers conditions were represented in the training dataset.

We found that the DNNs outperformed the linear model by a considerable margin when reconstructing the envelope of an attended speaker in competing-speaker scenarios, as well as in background babble noise. All three methods performed very similarly at the task of reconstructing the unattended speaker in the competing-speaker scenarios.

The three methods also performed very similarly at reconstructing the envelope of clean speech in foreign Dutch. The comprehension score in this listening condition was 0%, and it is has been shown that cortical speech tracking in the delta band is modulated by the speech comprehension level [16]. Since very low comprehension levels were not represented in the training data, this may explain why the DNNs did not perform as well in this listening condition.

4.5. Attention decoding performance

Finally, we decoded auditory attention in competing-speaker scenarios using the subject-specific decoders that were trained with Dataset 2. It was found that the use of DNNs was advantageous for this purpose, as was shown in [9]. We also replicated the finding that a short window length of about 2 s was optimal for real-time applications, in the sense that the information transfer rate (ITR) defined in equation (2) is maximised. We note that this ITR does not account for the total delay required by the proposed decoding algorithm: EEG filtering operations and audio processing operations have been neglected. The number of samples used to calculate the correlation coefficients as well as the number of temporal input samples T have been included. The bitrates that were achieved by the DNNs in this work were somewhat lower than those reported in [9], and the differences cannot be fully explained by the fact that we accounted for the temporal receptive field duration T in our calculation. The differences might be explained by the fact the authors trained their DNN using EEG recorded in a competing-speaker scenario, which was the same listening condition as was used for evaluation. Despite these differences, our study provides conclusive evidence that DNNs can be used for enhanced and robust decoding of selective attention in competing-speaker scenarios.

Acknowledgment

Michael Thornton was supported by the UKRI CDT in AI for Healthcare https://ai4health.io (Grant No. P/S023283/1).

Code availability

Supporting Python code is available at https://github.com/Mike-boop/mldecoders. This package contains all the functions used for data preprocessing, model training, and analysis.

ORCID iDs

Mike Thornton https://orcid.org/0000-0002-2235-5879
Danilo Mandic https://orcid.org/0000-0001-8432-3963
Tobias Reichenbach https://orcid.org/0000-0003-3367-3511

References

[1] Lesica N A 2018 Why do hearing aids fail to restore normal auditory perception? *Trends Neurosci.* **41** 174–85

[2] 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 2014 Attentional selection in a cocktail party environment can be decoded from single-trial EEG *Cereb. Cortex* **25** 1697–706

[3] 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 *Front. Neurosci.* **12** 262

[4] Looney D, Park C, Xia Y, Kidmose P, Ungstrup M and Mandic D P 2010 Towards estimating selective auditory attention from EEG using a novel time-frequency-synchronisation framework *Proc. 2010 Int. Joint Conf. on Neural Networks (IJCNN)* pp 1–5

[5] Bleichner M G, Mirkovic B and Debener S 2016 Identifying auditory attention with ear-EEG: cEEGrid versus high-density cap-EEG comparison *J. Neural Eng.* **13** 066004

[6] Fiedler L, Wöstmann M, Graversen C, Brandmeyer A, Lunner T and Obleser J 2017 Single-channel in-ear-EEG detects the focus of auditory attention to concurrent tone streams and mixed speech *J. Neural Eng.* **14** 036020

[7] Forte A E, Etard O and Reichenbach T 2017 The human auditory brainstem response to running speech reveals a subcortical mechanism for selective attention *eLife* **6** e27203

[8] Etard O, Kegler M, Braiman C, Forte A E and Reichenbach T 2019 Decoding of selective attention to continuous speech from the human auditory brainstem response *NeuroImage* **200** 1–11

[9] de Taillez T, Kollmeier B and Meyer B T 2020 Machine learning for decoding listeners' attention from electroencephalography evoked by continuous speech *Eur. J. Neurosci.* **51** 1234–41

[10] Ciccarelli G, Nolan M, Perricone J, Calamia P T, Haro S, O'Sullivan J, Mesgarani N, Quatieri T F and Smalt C J 2019 Comparison of two-talker attention decoding from EEG with nonlinear neural networks and linear methods *Sci. Rep.* **9** 11538

[11] Mandic D P and Chambers J A 2001 *Recurrent Neural Networks for Prediction* (New York: Wiley)

[12] de Taillez T, Denk F, Mirkovic B, Kollmeier B and Meyer B T 2019 Modeling nonlinear transfer functions from speech envelopes to encephalography with neural networks *Int. J. Psychol. Stud.* **11** 1

[13] Geirnaert S, Vandecappelle S, Alickovic E, de Cheveigne A, Lalor E, Meyer B T, Miran S, Francart T and Bertrand A 2021 Electroencephalography-based auditory attention decoding: toward neurosteered hearing devices *IEEE Signal Process. Mag.* **38** 89–102

[14] Lawhern V J, Solon A J, Waytowich N R, Gordon S M, Hung C P and Lance B J 2018 EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces *J. Neural Eng.* **15** 056013

[15] Weissbart H, Kandylaki K D and Reichenbach T 2020 Cortical tracking of surprisal during continuous speech comprehension *J. Cogn. Neurosci.* **32** 155–66

[16] Etard O and Reichenbach T 2019 Neural speech tracking in the theta and in the delta frequency band differentially encode clarity and comprehension of speech in noise *J. Neurosci.* **39** 5750–9

[17] Gramfort A *et al* 2013 MEG and EEG data analysis with MNE-Python *Front. Neurosci.* **7** 1–13

[18] Hastie T, Tibshirani R and Friedman J 2001 *The Elements of Statistical Learning* (*Springer Series in Statistics*) (Berlin: Springer)

[19] Bishop C M 2006 *Pattern Recognition and Machine Learning* (*Information Science and Statistics*) (Berlin: Springer)

[20] Schmidhuber J 2015 Deep learning in neural networks: an overview *Neural Netw.* **61** 85–117

[21] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 770–8

[22] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58 (available at: http://jmlr.org/papers/v15/srivastava14a.html)

[23] Clevert D-A, Unterthiner T and Hochreiter S 2016 Fast and accurate deep network learning by exponential linear *4th Int. Conf. on Learning Representations* (*San Juan, Puerto Rico, 2–4 May 2016*) (arxiv:1511.07289)

[24] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *Proc. 32nd Int. Conf. on Machine Learning (ICML'15)* vol 37 pp 448–56

[25] Santurkar S, Tsipras D, Ilyas A and Mądry A 2018 How does batch normalization help optimization? *Proc. 32nd Int. Conf. on Neural Information Processing Systems* pp 2488–98

[26] Tompson J, Goroshin R, Jain A, LeCun Y and Bregler C 2015 Efficient object localization using convolutional networks *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 648–56

[27] Dozat T 2016 Incorporating Nesterov momentum into Adam *Proc. Int. Conf. on Learning Representations (ICLR) Workshop*

[28] McFarland D J, Sarnacki W A and Wolpaw J R 2003 Brain–computer interface (BCI) operation: optimizing information transfer rates *Biol. Psychol.* **63** 237–51

[29] Wolpaw J, Ramoser H, McFarland D and Pfurtscheller G 1998 EEG-based communication: improved accuracy by response verification *IEEE Trans. Rehabil. Eng.* **6** 326–33

[30] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32 (Curran Associates, Inc.) pp 8024–35

[31] Virtanen P *et al* (SciPy 10 Contributors) 2020 SciPy 1.0: fundamental algorithms for scientific computing in python *Nat. Methods* **17** 261–72

[32] Seabold S and Perktold J 2010 Statsmodels: econometric and statistical modeling with python *Proc. 9th Python in Science Conf.* pp 92–96

[33] Masters D and Luschi C 2018 Revisiting small batch training for deep neural networks (arXiv:1804.07612)

[34] Smith S, Elsen E and De S 2020 On the generalization benefit of noise in stochastic gradient descent *Proc. of the 37th Int. Conf. on Machine Learning* (PMLR) pp 9058–67

[35] Zhu M and Gupta S 2017 To prune, or not to prune: exploring the efficacy of pruning for model compression (arXiv:1710.01878)

[36] Frankle J and Carbin M 2019 The lottery ticket hypothesis: finding sparse, trainable neural networks *Proc. 7th Int. Conf. on Learning Representations (ICLR 2019)*

[37] Bronstein M M, Bruna J, Cohen T and Veličković P 2021 Geometric deep learning: grids, groups, graphs, geodesics, and gauges (arXiv:2104.13478)

[38] Aroudi A, de Taillez T and Doclo S 2020 Improving auditory attention decoding performance of linear and non-linear methods using state-space model *2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* pp 8703–7

[39] Fuglsang S A, Dau T and Hjortkjær J 2017 Noise-robust cortical tracking of attended speech in real-world acoustic scenes *NeuroImage* **156** 435–44