Auditory Attention Classification with Contrastive Learning

Gautam Sridhar

Sofía Boselli

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6235
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2024

Abstract

Auditory attention detection is crucial for understanding speech in noisy environments, a challenge known as the "cocktail party problem." This project investigates the use of electroencephalography (EEG) to identify which speaker a listener attends to. EEG's portability and real-time recording capabilities make it a promising tool for practical applications.

We propose a novel neural network model for auditory attention detection using EEG data. The model reconstructs the attended speech envelope while simultaneously classifying attended vs. unattended speech. It incorporates a contrastive learning loss function (SigLIP), which, to our knowledge, has not been previously applied to EEG-based auditory attention detection. The model architecture combines convolutional, fully connected, and attention layers.

Evaluated on an EEG dataset with 31 subjects, the model achieves a mean accuracy of 68% and a mean correlation of 0.105 between the reconstructed and attended envelopes. This surpasses the baseline performance of linear methods (63% accuracy, 0.084 correlation). These results suggest the potential of contrastive learning for improving auditory attention detection accuracy, warranting further investigation.

Acknowledgements

We would like to thank our supervisors Bo Bernhardsson, Emina Alickovic and Martin Skoglund as they have helped us every step of the way, giving us feedback and helping us guidance on structuring the project.We would also like to thank Eriksholm Research Centre for giving us the opportunity to work on this project and for providing the necessary datasets. Lastly, thank you to the National Academic Infrastructure for Supercomputing in Sweden (NAISS) project for making computational resources available under the project NAISS2024/22-283.

Contents

1

Introduction

1.1 The Problem with Auditory Attention

We often find ourselves listening to one person while there is background sound, this being music, ambient noises or even the conversations of others. For humans, effortlessly focusing on the desired speaker is a natural ability. However, for computers, this "cocktail party problem" poses a significant challenge. This project investigates this problem, with a simple example presented in Figure 1.1. Solving it could have immense benefits in many areas, especially in development of hearing. Current hearing aids struggle to identify and isolate the relevant sound for the hearing aid user in noisy environments, leading to listening discomfort and social withdrawal [Marrone et al., 2008].

This project focuses on identifying the attended speaker by analyzing the brain signals of the listener through EEG. Research in this area has grown significantly in recent years, employing both linear and non-linear methods (as outlined in the following sections). EEG's portability and ease of use, requiring a few strategically

Figure 1.1 Cocktail party problem

placed sensors - such as those placed along the ear area of the user - make it a viable option for real-world, live attention tracking.

1.2 Objectives

The proposed objectives for this project are:

- Apply contrastive learning to the cocktail party problem.
- Analyze whether contrastive learning presents an advantage in this context.
- Obtain an accuracy higher than the chosen baseline method.
- Aim to improve accuracy further than existing attention decoding models.
- Obtain a high correlation between original and reconstructed stimuli.

2

Background

2.1 Approaches to the problem

There are multiple ways to approach the cocktail party problem. One method that has gained popularity with the advent of deep learning is to classify the locus of attention as a binary classification task. Another approach is to reconstruct the stimulus from the EEG recordings, which can be done with both linear and non-linear models. Once the stimulus is reconstructed, the correlation with the stimuli is calculated and the one which scores higher is defined as the attended one. This is referred to as the **backwards approach**, where the EEG is used to reconstruct the speech stimulus. The opposite process, known as the **forward approach**, entails reconstructing the EEG signal from the speech stimuli. In this project, we employ the backward approach.

Linear methods

The linear methods that are capable of transforming an EEG recording into a sound envelope used in this project are described in [Alickovic et al., 2019]. These methods rely on Finite Impulse Response (FIR) models to estimate a Temporal Response Function (TRF) [which essentially captures how the speech signal influences the EEG signal at different time lags]. Given an EEG signal $X_i \in \mathbb{R}^{C \times T}$ (where C is the number of channels, T is the number of samples, and i is the corresponding segment index), we can reconstruct an estimate of the attended speech envelope $s_i(t)$ as,

$$\hat{s}_i(t,\theta) = \sum_c \sum_l X_i(t+l,c)\theta(l,c).$$

Here, θ represents the estimated TRF, which acts as a linear mapping from the EEG onto the speech stimulus. The TRF θ is estimated by minimizing the Mean Squared Error (MSE) between the actual speech envelope and the reconstructed envelope. In this project, the backward model was implemented using the mTRF toolbox [Crosse et al., 2016a] which tries to minimize the MSE between the estimated envelope and the attended speech envelope. It uses ridge regression, which adds a regularisation

term λ, as trying to only minimise the MSE would cause the model to overfit on the training data. The formula for ridge regression is

$$\hat{\theta} = \arg\min_{\theta} \left(\|\hat{s}_i(t,\theta) - s_i(t)\|_2^2 + \lambda \|\theta\|_2^2 \right).$$

There are multiple ways to extract the envelope of a speech signal, as outlined in [Biesmans et al., 2017]. This project utilizes the simple yet effective method of calculating the absolute value of the Hilbert transform of the speech signal, as done in [Alickovic et al., 2019]

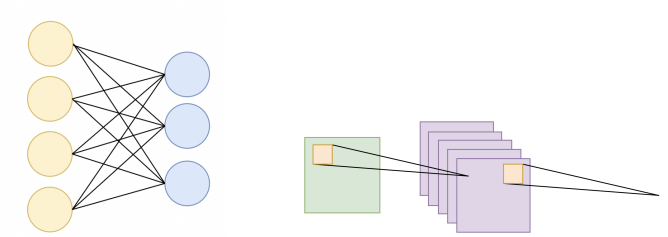
Non-Linear Methods

Neural networks offer an alternative approach to solving the same problems using non-linear models. These models are typically larger and more complex compared to linear methods.

Two popular neural networks are the Fully Connected Neural Network (FCNN) and the Convolutional Neural Network (CNN). The fully connected layer is simpler, where every node in the layer connects to all nodes in the previous layer, justifying its name. This is mathematically described by the equation:

$$y(x) = f(Wx + b).$$

Here, x is the input vector, W represent the weights, b is the bias and f is the non-linear activation function (e.g., Sigmoid, Rectified Linear Unit (ReLU), Hyperbolic Tangent Function). A visualization is provided in Figure 2.1a.



(a) Fully Connected Layer

(b) Convolutional Layer

Figure 2.1 Popular Neural Network Layers

A convolutional layer is more commonly used for image data, but it can be applied to other types of inputs like EEG signals. In this case, the kernel 'slides' through the input, extracting features relevant to the task. A visualization of these layers is shown in Figure 2.1b.

Large neural networks typically combine different layers depending on the type of input data and the desired result. Once the architecture is defined, the network undergoes training. This involves adjusting the weights of the network to obtain the optimal results (e.g., classification, reconstruction). Training relies on a loss function (measurement of how close the current network is to the desired function) and an optimizer (updating the weights based on the selected loss).

These models can be used for two main purposes: predicting the envelope or simply classifying audio inputs as attended or unattended. In the first option the model predicts clear and defined speech features (this is envelope, spectrogram etc.), while in the second option the features are implicitly chosen and analyzed by the network. While the latter approach avoids explicit feature selection, it can be susceptible to overfitting. Deep neural networks may exploit patterns in the data that are not relevant to the task, as shown in [Puffay et al., 2023]. For example, the network might overfit to trial-specific details or even people's eye gaze patterns during the recordings. In contrast, estimating the stimulus directly from the EEG data avoids these potential pitfalls.

2.2 Previous methods

There are two main approaches for using neural networks to understand auditory attention with EEG:

- Stimulus reconstruction (SR): This approach aims to reconstruct the envelope of the attended speech from EEG.
- Locus of Attention (LoA): This approach directly classifies which sound source the listener is attending to from EEG.

Both methods have been investigated in recent years. A study by [Thornton et al., 2022] compared two approaches: one using solely Feed-Forward Fully Connected Layers and another using CNNs. Both networks took the EEG signal as input and output the value of a sample of the envelope at a given time. The final result was then correlated with the original attended envelope. The fully connected network comprised multiple fully connected layers with the hyperbolic tangent as activation function and dropout for regularization. The CNN utilized convolutional layers, batch normalization layers, average pooling, dropout and again hyperbolic tangent as activation function. The loss was calculated as the negative correlation coefficient between the reconstructed and the real envelope, and Adam was used as an optimizer. Thornton's study achieved a mean reconstruction score (Pearson correlation) of around 0.14 for the fully connected network and 0.16 for the convolutional network, using a 3-second window and subject-specific models on a 13-participant

dataset. Attention decoding with this model yielded the accuracies of $\sim$ 65% (2-second window) and 72% (5-second window). Subject-independent models (one subject left out for testing) resulted in a mean score of approximately 0.11.

In [Vandecappelle et al., 2021], a CNN was used to decode the LoA. The CNN takes the EEG signal as input and outputs two values, indicating whether the listener was attending the sound on the left or right. The network architecture consists of convolutional layers with ReLU as activation function, an average pooling layer, and a final fully connected layer for classification. The loss function used to train the network is the cross-entropy loss between the predicted and real labels, and the chosen optimization is mini-batch stochastic gradient descent. The study achieved a median accuracy of 85.1% for a 10-second window and 80.8% for a window of 1-second window on a dataset of 16 normal-hearing subjects using subject-specific training. However, the accuracy dropped to 69.3% for a 1-second window using for subject-independent training where (one subject left out).

As seen in the previous examples, there are two main approaches to analyzing model results based on how subject data is handled: subject-specific and subject-independent results.

- **Subject-specific model:** In this approach the model is trained, validated and tested with data of the same subject. This is valuable as it can provide information on how good the model performs by analyzing the brain information of one specific person. As brain waves are specific to each person, the same hyperparameters and weights could be beneficial to one subject and damaging to another, so a greater accuracy can be achieved in this manner. It can also be trained and tested on multiple subjects to take a wider selection of readings.

- **Subject-independent model:** Here, the model is trained on data from a group of subjects and then tested on unseen data from different subjects. This approach is more practical for real-world applications, as training on every individual user wouldn't be feasible. As mentioned before, the EEG readings can be very different from subject to subject, which means that this result will probably be considerably lower.

2.3 Convolutional Neural Networks

CNNs have been shown to yield noticeably better results for various tasks compared to simple FCNNs. Here, 2D CNNs are used mainly for image data. To address this, several methods have been developed to convert EEG data into an image representation that can be fed to the neural network. For example, in [Lawhern et al., 2016] and [Thornton et al., 2022], the authors use 2D convolutions by transforming the EEG data into an image (with spatial and temporal features), which are then passed

through convolutional layers. While 2D Convolutions can handle both spatial and temporal features, we opted to implement a 1D CNN that focused solely on temporal features.

2.4 Attention

In 2017, [Vaswani et al., 2017] introduced the transformer model and the multi-head self-attention mechanism. This machine learning architecture was initially proposed for Natural Language Processing applications and works in a sequence-to-sequence manner, meaning it transforms one sequence into another.

Attention is a way to calculate weights that reflect the importance of each datapoint both individually and as part of a sequence. In the context of language processing, this means that relevant words in a sentence received higher attention values, incorporating context into the model. The attention mechanism proposed in the paper is called Scaled Dot Product Attention and is obtained by calculating three matrices known as Queries, Keys and Values, which are then combined as follows:

$$\text{Attention} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

with d_k being the dimension of the keys. This method is expanded into Multi-Head Attention, where the attention is calculated in parallel by multiple heads and then concatenated. This is shown in Figure 2.2.

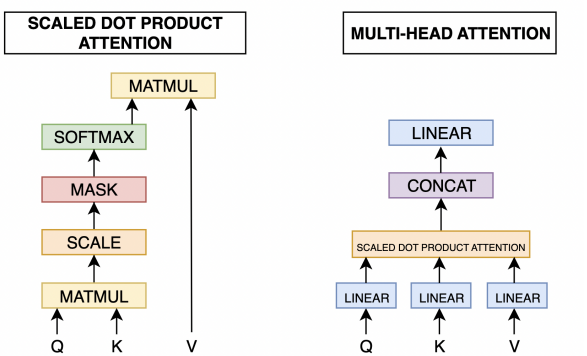


Figure 2.2 Attention Mechanism and Multi-Head Attention (Image inspired from [Vaswani et al., 2017])

2.5 Contrastive Learning

Overview on Contrastive Learning

Contrastive learning is a machine learning technique for training models in which the inputs are transformed into a representation vector and then compared with other representations in order to classify similar inputs together. It is said to be contrastive, as the training is done in a way that minimizes the distance between similar inputs and maximizes the distance with the rest. It can be applied in both supervised and unsupervised scenarios. It is one of the most popular methods in unsupervised settings, as it can achieve good results without requiring large amounts of labeled data. We chose to use contrastive learning as it seemed intuitive. The model requires us to "ignore" the masked speech and "focus" on the attended speech. Thus there is a clear positive and negative sample, and the model can now learn to predict the attended speech features.

CLIP and its variants

The Contrastive Language-Image Pretraining model (CLIP) [Radford et al., 2021] by OpenAI is a notable example of contrastive learning applied to image classification. CLIP learns by attempting to predict the textual description that best matches the image. One key aspect of CLIP is its loss function, which applies contrastive learning techniques. The CLIP loss works by projecting multimodal inputs onto similar representations while ensuring dissimilar representations are farther apart. Figure 2.3, adopted from the original paper, illustrates this process. In this figure, a batch of training data is chosen, and the representation of each element is obtained. The inner product between the representation of each image and each label is calculated, resulting in a matrix. The loss function is designed to maximize the dot product between the projections of similar views of data and minimize the projection of dissimilar views. It does so by using a softmax loss and setting the labels as the diagonals of the resultant matrix.
One benefit of CLIP is that it avoids the need for explicitly creating negative instances during training, unlike some other contrastive methods. Instead, it efficiently compares a view in a minibatch with all the other views in the minibatch. This makes the batch size a highly important hyperparameter.

For this project, a variant of CLIP known as SigLIP [Zhai et al., 2023] is used. It replaces the softmax loss in CLIP with a sigmoid loss instead. Due to CLIPs softmax loss, the loss of a positive pair depends on all the negative pairs in the data (a positive pair refers to a pair of datapoints that belong together, e.g. image with correct label, a negative pair would be an image with the wrong label). However, for SigLIP, the loss (both for negative and positive pairs) are independent of other examples in the minibatch. This essentially transforms the problem from a multiclass classification, to a binary classification problem. SigLIP has also shown to outperform CLIP on smaller batch sizes. An algorithm for SigLIP is shown in Algorithm 1.

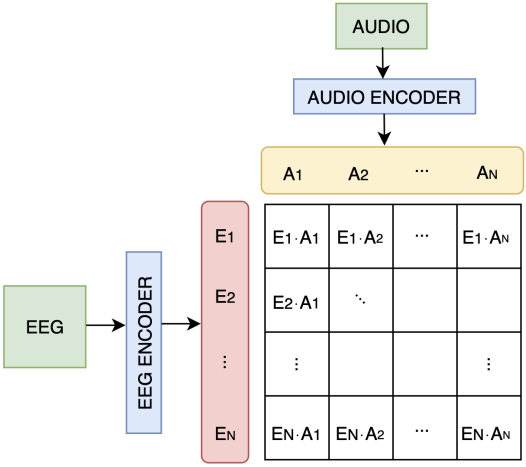


Figure 2.3 CLIP Loss (Image inspired from [Radford et al., 2021])

Algorithm 1: SigLIP loss

```
# eeg = EEG embedding (batch size,num_channels,length)
# aud = Audio embedding (batch size,num_channels,length)
# b = bias
# t' = temperature
n = eeg.shape[0]
t = exp(t')
eeg = normalize(eeg)
aud = normalize(aud)
logits = einsum('btc','rtc'->'br',eeg,aud)*t + b
labels = 2*eye(n) - ones(n)
loss = -sum(logsigmoid(logits*labels))/n
```

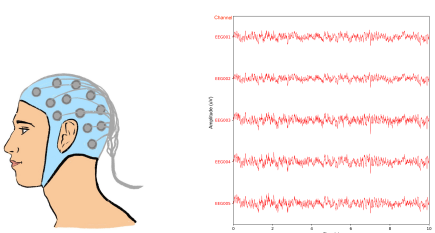
3

Dataset

The dataset [Alickovic et al., 2021] used in this project was provided by Eriksholm Research Centre. The dataset consists of EEG data measured from human participants along with the corresponding speech stimuli they listened to during the experiment. The following sections provide a detailed description of the dataset.

3.1 EEG and Audio

Electroencephalography (EEG) is a method for recording electrical activity generated by the brain. It involves placing sensors at a specific scalp location to capture this activity. The number of sensors used may vary depending on the desired information. EEG is a valuable tool in various fields, including neuroscience and cognitive research. An example of an EEG measuring is shown in Figure 3.1.

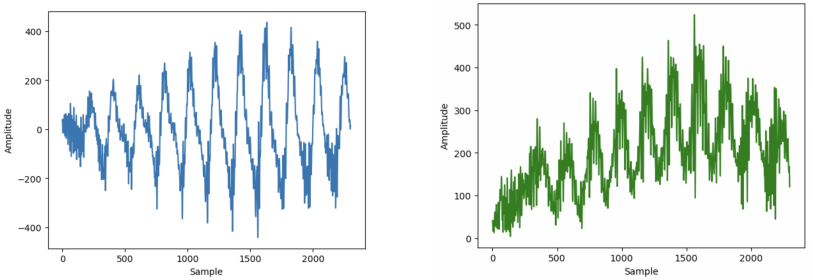


(a) Sketch of sensor cap used to measure EEG

(b) Example of an EEG reading

Figure 3.1 EEG measuring (5 channels of 64 shown)

The audio is obtained as a common sound recording and the different speech features such as envelope, spectrogram, phonemes etc. can be calculated as in Figure 3.2.



(a) Audio recording **(b)** Envelope of recording

Figure 3.2 Audio data

3.2 Experimental Design

The dataset consisted of 34 native danish speakers as participants, with ages ranging from 21 to 84 (mean 64.2 and std 13.6). All participants had mild to moderately severe symmetrical sensorineural hearing loss and were experienced hearing aid users. Additionally, they reported no history of neurological disorders, dyslexia or diabetes mellitus.

3.3 Setup

The experiment involved recording EEG data from participants while they listened to two competing talkers. Participants were instructed to attend only one of these talkers and completely ignore the other one. A simple diagram of this setup is shown in Figure 3.3.

EEG data were recorded using a BioSemi ActiveTwo recording system (Amsterdam, Netherlands). with a sampling frequency of 1024Hz. The system used 66 electrodes, with 2 reference electrodes placed on the mastoids. The experiment took place in a soundproof room where participants sat facing speakers positioned at $\pm 30°$, $\pm 112.5°$,$\pm 157.5°$ azimuth relative to them. The two front loudspeakers ($\pm 30°$) were used for the attended and the ignored speech, while the four speakers behind the participants ($\pm 112.5°$ and $\pm 157.5°$) played babble noise to increase listening difficulty. The speech stimuli consisted of news clips of neutral content to minimize any emotional responsea. All silences longer than 200ms were trimmed, so as to not exceed 200ms. A 3dB Sound Pressure Level (SPL) between the attended speech and the babble was maintained. Each trial started with 5 seconds of background noise followed by 33 seconds of speech stimuli. After 38 seconds, the participants were asked a question about the content of the attended speech. The experiment consisted of 80 trials in total, presented in 4 blocks of 20 trials each. Each block used a different hearing aid noise reduction setting.

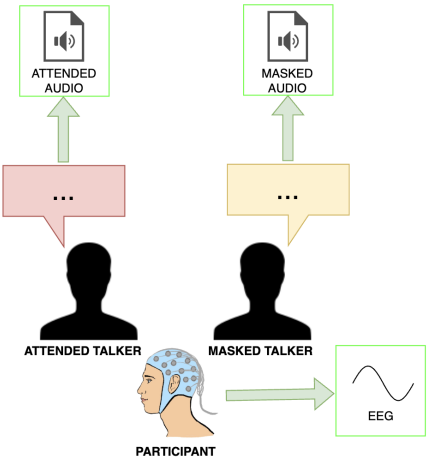


Figure 3.3 Experiment setup and obtained data

3.4 Initial Preprocessing

The data provided had already been pre-processed to remove noise and other artefacts such as eye-blinks, muscle movement, heartbeats, powerline noise etc. The preprocessing steps were as follows

1. Bandpass filtering between 0.5Hz and 70Hz using a zero-phase Hamming window FIR.
2. Narrow band Notch filter (49Hz-51Hz) to remove line noise.
3. Downsampling to 256Hz.
4. Removing contaminated EEG channels manually, and replacing them with data interpolated from surrounding clean EEG channels.
5. Further removing residual artifacts such as eye movements, eye blinks, muscle activity, heart beats and single channel noise using ICA.

After performing the above pre-processing, only 31 out of the inital 34 speakers were retained. This was due to there being persistent artefacts in the EEG data, along with other issues. For subjects 21-24, a corrupted block of trials was also removed.

3.5 Pre-Processing

The models were implemented using PyTorch [Ansel et al., 2024] as the base, with PyTorch Lightning [Falcon and The PyTorch Lightning team, 2019] to make the

training easier to debug and visualize. Each trial was converted from its original format (.mat) into a more Python-friendly format (.npz) for easier access and analysis. First, the EEG data was bandpass filtered between 1Hz and 16Hz using a 3rd order Butterworth filter. It was then downsampled to 64Hz. The audio data for the attended and the masker speech is preprocessed in 2 ways. One method involved calculating the envelope of the speech signal. To calculate the envelope, the audio data is downsampled to 16kHz and then the absolute value of the Hilbert transform is computed. This is then downsampled to 64Hz. The other method involved creating a melspectrogram, a representation that captures both frequency and time information of the audio. Here, the number of mels was set to 32, the number of ffts was set to 512, and the hop length was 250 samples. This resulted in a 64Hz melspectrogram. Finally, the melspectrogram and the envelope were concatenated together. This resulted in the final shape of the audio data of $(33, 2112)$. The shape for the EEG data is $(64, 2112)$. The audio preprocessing was done using the python package Librosa [McFee et al., 2024].

3.6 Dataset Split

Splitting the data into training, validation, and test set needed to be handled very carefully, as leakage of training data into validation or test set would contaminate the results giving inflated performance [Puffay et al., 2023; Tanveer et al., 2024]. One common method was to split every trial, ensuring that the windows in the training set were not close to those in the validation or test sets. However, this approach was deemed suboptimal, as the model might learn interconnections within trials rather than connections between EEG and stimulus. Therefore, we decided to split the data into non-overlapping trials, ensuring each set (training, test, validation) had a different trial to prevent data leakage. The dataset also had to be split evenly, as there are 4 different blocks of trials, with each having their own hearing aid noise reduction setting. Initial tests with the data revealed that the blocks with noise reduction turned 'on' were generally easier for the model to work with. Therefore, we grouped the blocks based on noise reduction status ('on' or 'off'). Within each group, we further split the data into train, validation and test sets, ensuring that the trials were randomized but also ensuring each had an equal amount of trials with noise reduction 'on' and 'off'. This ensures the model is not biased towards either noise reduction setting.

4

Methodology

4.1 Baseline

A common approach to assess model performance is to compare it to a simpler baseline model. In this project, we use a backward linear model similar to the one described in [Alickovic et al., 2019] (i.e. one with EEG as input and audio envelope as output). To implement this baseline model, we used the mTRF-Toolbox for Matlab presented in [Crosse et al., 2016b]. We set the minimum and maximum time lags to -100ms and 400ms from "onset", respectively. The time lag allows for the possibility that the EEG and audio signals are not aligned in time, a diagram is shown in Figure 4.1. A regularization value of $\lambda = 10^5$ was used. The reconstructed envelope was then correlated with both the attended and masked envelope to assess classification accuracy.

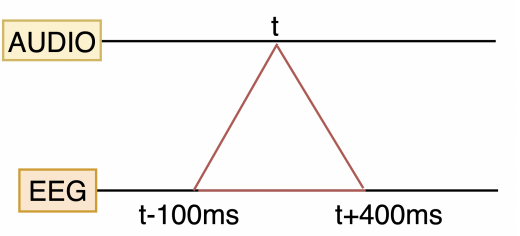


Figure 4.1 An example of the time lag option

The model was trained and tested in a subject-wise manner. The model was trained with the training portion of a subject and tested with the remaining trials.

4.2 Proposed Architecture

The objective of the proposed model is mainly to achieve a high accuracy when classifying speech signals as attended or unattended. As a secondary objective, the

goal is to also obtain a high correlation between the original attended envelope and the reconstruction obtained from the EEG. Relevent code for the architecture as well as the loss implementation can be found on this projects Github [Boselli and Sridhar, n.d.] The proposed architecture was kept simple, as the project mainly focused on exploring how contrastive learning affects the model. While transformers were initially implemented, the architecture proved to be too complex and the model would almost certainly overfit. Therefore, the architecture was simplified to a base state. We proposed a simple convolutional model along with attention layers and skip connections

The proposed model is composed of three main submodels as shown in Figure 4.2. The EEG, the attended audio and the ignored audio are encoded into a desired matrix of embedding size in two separate encoders. The EEG enters the encoder with the shape (B,EC,W) where B is the batch size which is a hyperparameter, EC the number of channels, in this case 64 and W the size of the window which can be calculated as $W = f_s \times WindowSize$. For example, with a sampling of 64Hz and a Window Size of 3, $W = 192$ samples. The audio signals enter the encoder with a shape of (B,AC,W) with AC being the number of speech features, in this case 33 (32 corresponding to the spectrogram and 1 to the envelope).

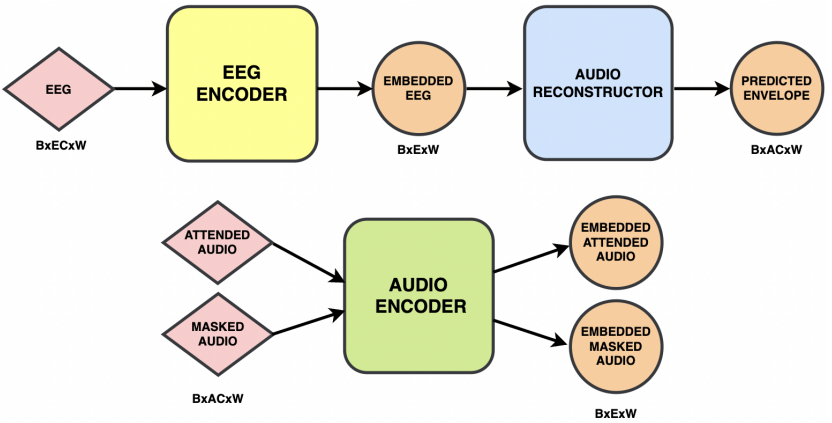


Figure 4.2 Diagram of model. B = Batch Size, EC = Number of EEG Channels, W = Window Length, AC = Number of Audio Channels, E = Size of Embedding

Both encoders result in an embedding of size (B,E,W) with E being the chosen embedding size which is also a hyperparameter. The EEG embedding is then fed into another block which reconstructs the envelope of the attended signal.

EEG Encoder

The EEG Encoder is shown in Figure 4.3. The EEG is first passed through a subject layer. This is a layer that takes into account the subject being analyzed in order to

obtain better results for each specific subject, this means it gathers specific weights for each subject and trains on them. As each subject can present different patterns in their EEG, this layer is created as an attempt to adapt the model for each subject independently. An example of its usage can be found on Figure 4.4.

The result from this subject layer is summed to the original EEG, this is then passed through a ECxE convolutional layer, kernel size 3, which takes the 64 channels and transforms them into the selected embedding size. This is then passed multiple times through a module that contains a ExE convolutional layer, a multi head attention layer of 3 heads, a dropout layer and a layer normalization layer. The K times this is done, the dropout value and the sizes of the convolutional kernel are passed to the network as hyperparameters. Finally, this is passed to a final ExE convolutional layer of kernel size 1 and the embedded EEG is obtained. The convolutional layers are chosen because of their ability to find useful features as well as to change the size of the input into the desired embedding size, dropout and normalization layers help to avoid overfitting. The attention layer is added in an attempt to capture information along the spatial aspect of the EEG and to provide an indication of if or what channels are more relevant when classifying and reconstructing the audio.

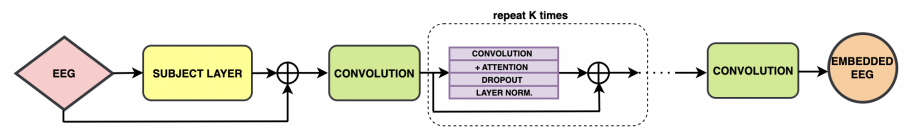


Figure 4.3 Diagram of the EEG Encoder

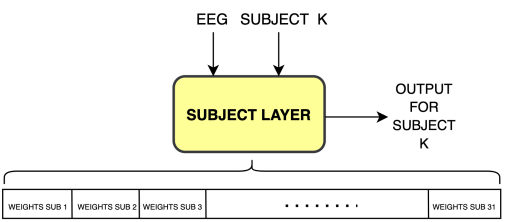


Figure 4.4 Subject Layer example, in this case the layer is given the EEG of subject K.

Audio Encoder

The audio encoder has a similar structure to the EEG encoder as shown in Figure 4.5. The audio is passed through a ReLU activation layer and a ACxE convolutional layer, kernel size 3. As before, this layer transforms the audio from its initial 33 channels into the desired embedded dimension. This is then passed multiple times through the ExE convolution-3 headed attention-dropout-layer normalization

combination. The final layer is a ExE convolutional layer of kernel size 1 from which the embedded audio is obtained.

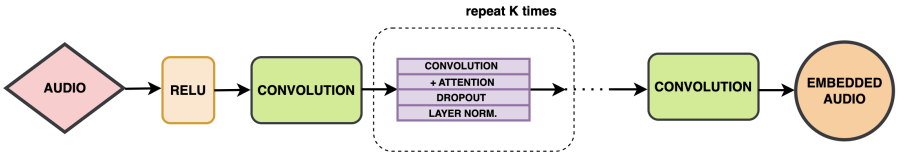


Figure 4.5 Diagram of the Audio Encoder

Speech Reconstruction

The speech reconstruction model is shown in Figure 4.6. This takes the embedded EEG obtained from the EEG encoder as input. The embedding is passed through a ExAC convolutional layer in order to obtain a result with the shape of the original audio data. This is introduced into the previously described combined block K times and passed through a ACxAC convolutional layer and a GeLU activation layer. This model returns a signal that should be the reconstruction of the attended speech.

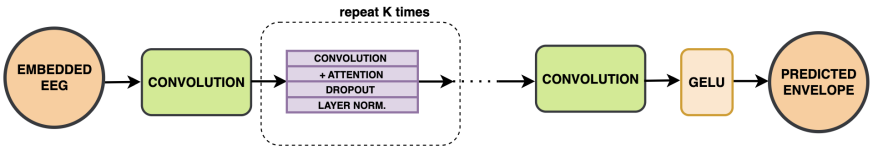


Figure 4.6 Diagram of the Audio Reconstructor

Training

The first step in training is to obtain the three embeddings, EEG, attended audio and masked audio by passing them through their respective encoders. The EEG embedding is then passed through the Reconstruction model and a reconstructed envelope is obtained.

The loss is calculated as:

$$L = L_{\text{contrastive}}(eeg\_emb, att\_emb) - L_{\text{Pearson}}(pred\_env, att\_env),$$

where *eeg_emb* is the EEG embedding obtained by passing the EEG data through the EEG encoder, and *att_emb* is the embedding obtained by passing the attended speech features through the Audio Encoder.

$L_{\text{contrastive}}$ is the SigLIP loss described before. An important aspect of the contrastive loss considered is the similarity metric used. The metric needs to be close to 1

for similar embeddings, and close to 0 for dissimilar ones, the metric also needs to be able to be calculated fairly quickly. Three different similarity metrics were considered, namely:

- An extension of the distance metric used in the SigLIP paper (dot product) for 2D embeddings
- Gaussian Kernel Distance implemented for 2D embeddings
- Pearson Correlation (rescaled to be between 0-1)

Dot Product. We extend the dot product to be able to take 2D inputs. This is essentially the summation of a hadamard product, however we require that the result be the function to be applied to every other example in the batch, and return a result in the form of a square matrix. We use the einsum function for this.

Gaussian Kernel Distance. The gaussian kernel distance can be used as a similarity metric, with the distance between two different embeddings being defined as

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right).$$

Here, the model had to optimize the parameter σ, which determines the width of the Gaussian kernel. This was implemented in the manner required for SigLIP (every embedding in the batch is compared with every other embedding).

Pearson Correlation. Pearson Correlation as a similarity metric between two embeddings was also considered. To compute the Pearson correlation between 2D inputs, we found the mean of the correlations along the channel axis between 2 examples. This was then re-scaled between 0-1.

While the Gaussian Kernel Distance converged faster, it was finally decided to use the dot product similarity measure, as it converged to the best validation correlation and accuracy.

During SigLIP loss the embeddings of every element in the batch are compared and contrasted. The same audio files were used for various subjects and trials which means that two elements in the batch could potentially have the same audio file as attended signal. This causes problems in the loss as the model will try to maximize the value with its own signal and minimize the value at the same time with the other. In order to avoid this a function was created that checks that no audio files are repeated in a batch, if this happens the datapoint is discarded from the batch and replaced randomly, this is done until the new datapoint is not already in the batch.

$L_{\text{Pearson}}(pred\_env, att\_env)$ is simply the Pearson correlation between the predicted envelope and the envelope of the original attended audio. This value should be as high as possible in order to obtain a good reconstruction.

Various metrics and combinations of them were tried as the loss but the one presented above was the one which yielded the best results. The optimizer used is Adamax with the learning rate and weight decay as hyperparameters. Early stopping on the validation loss is also implemented to avoid overfitting.

4.3 Statistical Testing

The statistical tests were performed in python using the library scipy [Virtanen et al., 2020]. In order to estimate the null set, the method outlined by [Crosse et al., 2021] was followed. The predicted envelopes were randomly cyclically shifted and then correlated with a random permutation of the true data. This procedure was performed 100 times in order to establish the null set. The significance level α was set to 0.05. Normality was checked, and a one-tailed unpaired Welch's t-test was used in order to calculate the significance of the obtained results. The results were first calculated over a population level and over each individual subject.

5

Results

All the results are shown with subject 14 left blank. This is because subject 14 was removed during the initial pre-processing. The final results were ran with 2 nodes with 8 Nvidia T4 GPUs each, the model was trained 4 times and averaged, this took approximately 3 hours.

5.1 Baseline

The linear model used as baseline was run for 3-second windows on data from 31 subjects. The results of attention classification accuracy per subject are shown in Figure 5.1. Overall, the model achieved a mean classification accuracy of 62.6% across all subjects. In terms of the Pearson correlations of reconstructed speech envelope with the envelopes of attended and ignored speech, the model achieved the mean Pearson correlations of 0.084 and 0.007 respectively. The results for each subject are presented in Figure 5.2. A comparison of both these correlations is also presented in Figure 5.3.

5.2 Proposed Architecture

Hyperparamter Tuning

The tool Ray presented in [Liaw et al., 2018] was used to carry out the hyperparameter search. The library takes the model, the parameters to tune and the possible values for each of them, then it runs the model multiple times with different combinations of the parameters and returns the option with the best results. The considered parameters and the final values are shown in Table 5.1.

Proposed Model Results

The model was trained and tested in windows of 3 seconds. On the test set a mean accuracy of 68% was obtained. Figure 5.4 outlines the accuracy obtained for each subject.

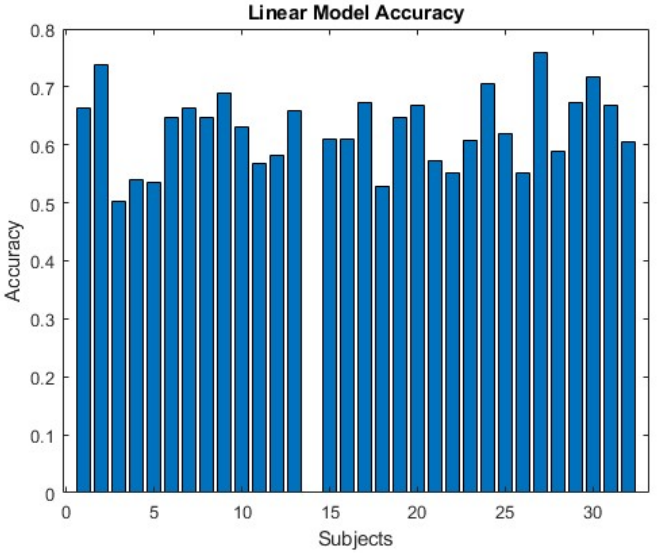
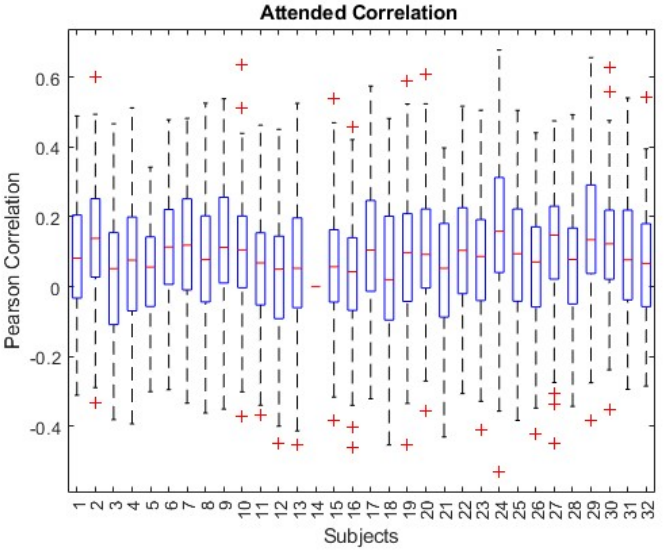


Figure 5.1 Mean classification accuracy per subject in baseline model. Subject number 14 is shown to be blank as the subject was removed during pre-processing.

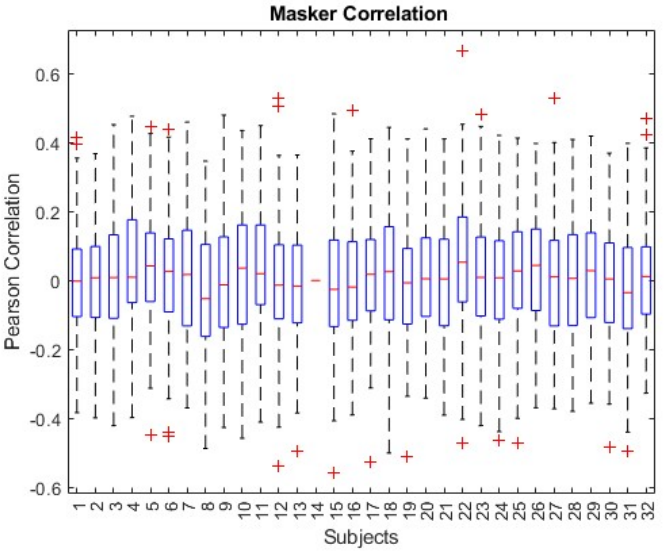
	Hyperparameter	Final Value
Data	Overlap	0.7
	Batch Size	128
Architecture	Kernels	[30,20,40]
	Temperature	10
	Bias	10
	Embedding	8
	Reconstruction Embedding	32
Regularization & others	Learning Rate	6e-4
	EEG Dropout	0.2
	Audio Dropout	0.4

Table 5.1 Final choice of hyperparameters. The hyperparameters were tuned with the Python Library Ray.Tune with which a grid search is conducted to obtain the best combination of parameters.

The mean correlation with the attended speaker is calculated as 0.105 while the mean correlation with the masked speaker is 0.017. The results per subject are shown in Figure 5.5. The differences between these two correlations was also calculated and plotted and can be seen in Figure 5.6.



(a) Attended speech correlation per subject (mean 0.084)



(b) Ignored speech correlation per subject (mean 0.007)

Figure 5.2 Boxplot of Pearson Correlations per subject in baseline model for attended and ignored speech. Outliers are represented by a cross outside the confidence intervals

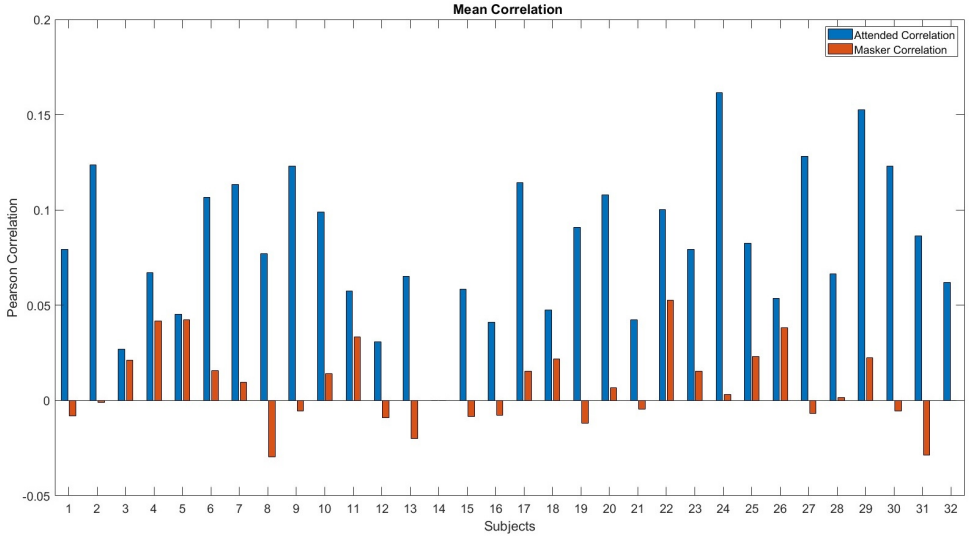


Figure 5.3 Comparison between attended and masked correlation in baseline model. The blue bars represent the correlation between reconstructed speech and the attended speech, while the red bars represent the correlation between reconstructed speech and the ignored speech.

5.3 Statistical Tests

We considered the null hypothesis to be that the Pearson Correlation obtained from the envelope reconstructed from the EEG is not significantly higher than the null set, which was designed by pairing cyclically shifted reconstructions with random true speech envelopes.The population level results, that is taking the entire dataset together in order to see if the model performs better on average, had the p-value $\lll \mathbf{0.05}$. When evaluated per subject in order to see if there is any significant improvement within the subject level, only subject number 21 and 26 had p-values greater than 0.05 (0.09 and 0.23 respectively), showing that the model is a performing significantly better than the null hypothesis.

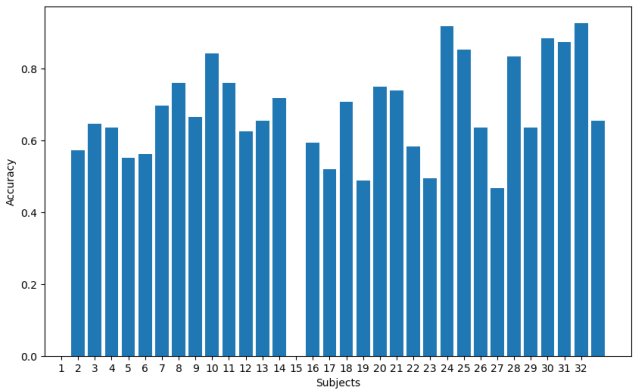
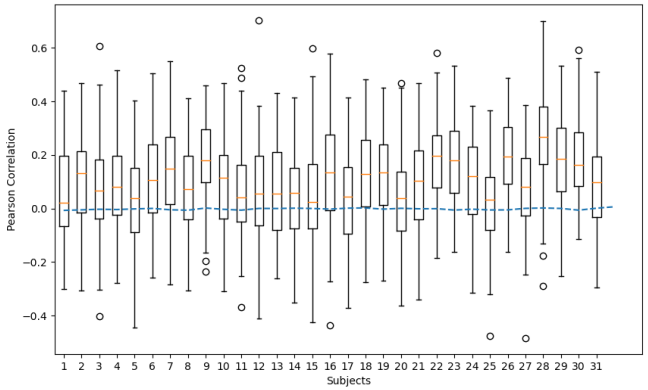
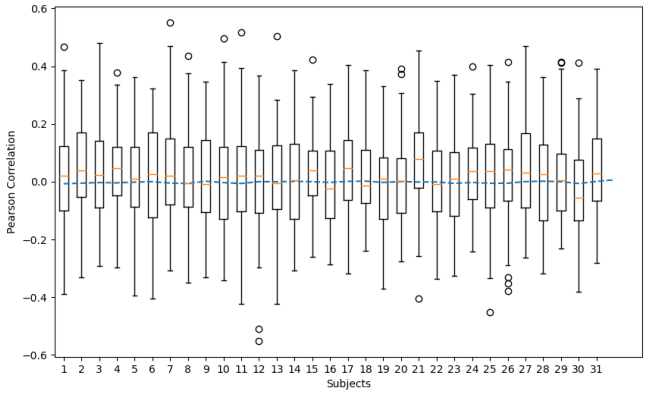


Figure 5.4 Accuracy per subject obtained using the model described in section 4.2 with parameters in table 5.1. As we can see, for most of the subjects the model performs better than random guessing, reaching 90% accuracy for a few cases as well. Subject number 14 was left blank as the subject was removed due to persistent artifacts.

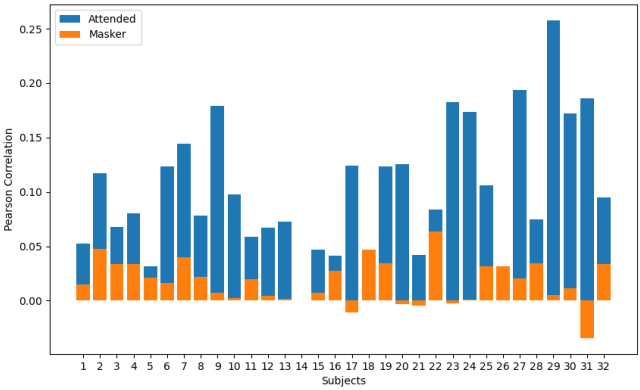


(a) Boxplot of the Attended Correlation per subject (dotted line indicates the median of the null set)

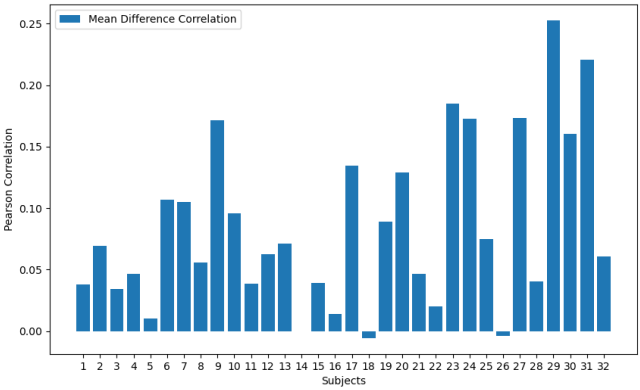


(b) Boxplot of the Masked Correlation per subject (dotted line indicates the median of the null set)

Figure 5.5 Correlation per subject on the test set, with a window size of 3 seconds. Outliers are represented as circles outside the confidence interval. The dotted line represents the median correlation of the null distribution, helping to highlight the model's ability to accurately reconstruct the attended speech envelope compared to chance. The mean correlation for the attended speech across subjects is significantly higher than that for the masked speech, indicating that the model can more accurately reconstruct the attended speech envelope



(a) Comparison of masked and attended correlation results on the test set, with a window size of 3 seconds. Each bar represents one subject, with subject 14 left blank. Attended Correlation is in Blue, and the Masker correlation is in orange



(b) Difference in correlation between masked and attended envelopes. Each bar represents one subject. The plot shows that as a whole, the reconstruction correlates much higher with the attended than the masker

Figure 5.6 Bar Plots to compare the pearson correlation of the reconstruction with Attended vs Masked.

6

Discussion

Progression of Loss and Accuracy during training

The accuracy across epochs is plotted in Figure 6.1. The model stops training due to early stopping, usually between 60 and 80 epochs, in this specific case after 72. While observing the validation loss and accuracy per epoch, it can be seen how the validation accuracy reaches its peak before the validation loss starts rising which causes the accuracy to decrease during a few epochs while the loss is still decreasing. This may be caused by the fact that the correlation with the attended audio keeps improving, but the correlation with the masked audio could be improving as well causing it to make wrong classifications thus decreasing the accuracy. The model could be optimized to prioritize only one of these metrics (correlation or accuracy) and this problem could potentially be avoided.

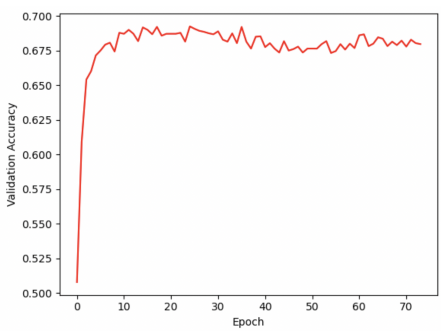


Figure 6.1 Accuracy variation per epoch

Although some subjects perform better than others it can be noted that the model always achieves accuracies above 50%.

Proposed Model vs Baseline

Our proposed model outperforms the baseline model in two key metrics:

- **Classification Accuracy:** The new architecture achieves an accuracy approximately 6% higher than the baseline. A comparison of the accuracy per subject between the proposed model and the baseline is shown in Figure 6.2.
- **Attended speech envelope reconstruction:** The model exhibits a stronger correlation (0.02 improvement) with the attended speech envelope compared to the baseline. The correlation obtained was around 0.1 which makes the 0.02 an improvement of 20%.

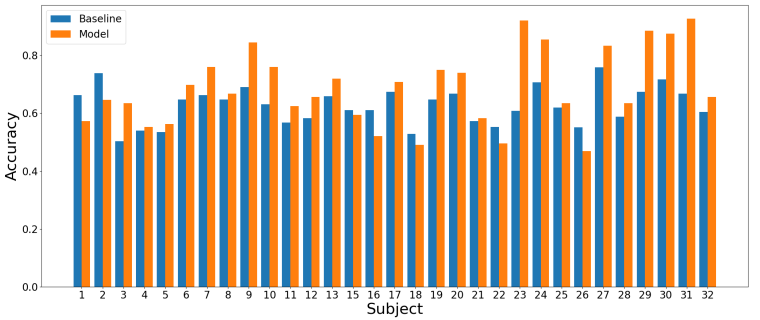


Figure 6.2 Comparison between accuracy per subject of proposed model and baseline. We can see that while the baseline performs better for some subjects, overall the model outperforms the baseline by a significant amount.

The ignored speech correlation is lower on the baseline which could imply that both correlations are more distinct (the correlations with the attended and unattended envelopes are further apart) in the baseline than on the proposed model which could impact the classification, but it does not seem to be a big issue as the accuracy is also improved.

It's important to note that while the average accuracy and correlation improve with our model, there might be individual subjects where the baseline performs better. This highlights the variability in EEG data and the need for further investigation.

Hyperparameter search

As previously mentioned, the hyperparameter search was conducted using Ray Tune, a tool that runs a specified number of trials by selecting different values from defined options and intervals. Due to constraints on time and resources, a complete grid search was not performed, which might have produced a better combination of parameters for the current architecture. The hyperparameters observed to be more influential to the final accuracy are overlap, batch size, embedding and kernels. The options and intervals of hyperparameters were chosen from sensible options, but in the case of the kernels the possible options can vary a lot, from the amount of kernels to the different sizes. A variety of kernels was added to the possible options and

these were changed according to the results obtained, but a lot of options remained to be analyzed.

Another possible issue is that the decrease in accuracy after a certain number of epochs (even as the validation loss keeps decreasing) could influence the search on choosing a combination that ended with a higher accuracy but had a lower peak.

Quality of dataset

It can be noted how the accuracy varies from one subject to another, as the success of the model is also very dependent on the quality of the data. For subjects where the attended correlation is very similar to the masked correlation, such as subjects 5 and 26, there exists the possibility that they were not fully attending the correct speaker during the experiment, which damages the accuracy results. It is also true that the brain signals captured by the EEG are very dependent on the subject, and that some subject-brains provide more information than others.

Advantage of Contrastive Loss

The accuracy is not only improved by maximizing the correlation with the attended audio but also by minimizing the correlation with the masked audio, this is achieved with the contrastive loss and it was noted that the accuracy decreased significantly if the SigLIP loss was not considered: when the model was run only considering the Pearson correlation as loss, this gave a mean accuracy of 62.8%, a mean attended correlation of 0.087 and a mean masked correlation of 0.013. The accuracy per subject is shown in Figure 6.3 and the correlation information is presented in Figures 6.4 and 6.5. The fact that the correlation is high is the result of using the Pearson correlation as loss, as the models sole objective will be to increase this correlation disregarding the accuracy. By adding the contrastive loss the model is forced to differentiate the attended correlation with the non-attended improving the accuracy.

Overfitting

As mentioned before the amount of data available is low compared to the optimal size of the datasets suitable for these kinds of models. The training data is even smaller as a part of the data being separated in validation and testing sets, this means that the training will suffer and the model easily overfits.

The model is simple in regard to the present architectures and depth, more complex structures were considered, but this caused the model to overfit faster which is why the current structure was maintained.

Other models

A comparison between the model presented in this project and other existing methods is difficult, as the results can be very different depending on the quality of the

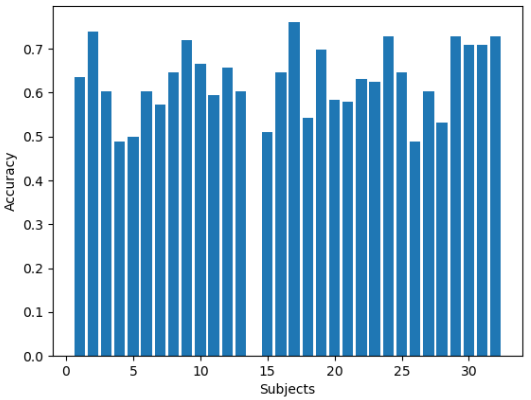
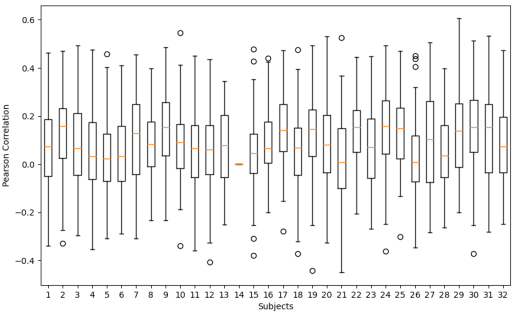


Figure 6.3 Mean Accuracy per subject when using only Pearson Correlation as loss. The accuracy per subject is lower than if we used contrastive loss, and we can see that the maximum accuracy any subject reaches is around 72%, which is much lower than if we implemented contrastive loss.

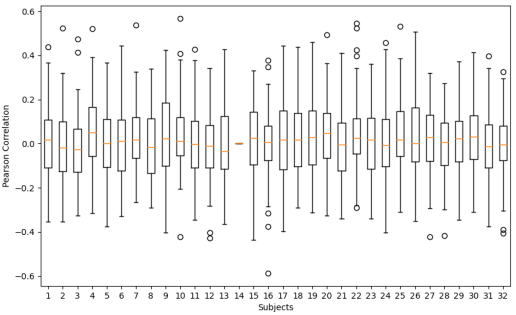
data. One key limitation is the privacy concerns associated with sharing the EEG dataset used in this project. This prevents from directly benchmarking models developed in previous studies on the same data. Another factor to consider is the specific population studied. Since our participants had hearing impairments, their EEG signals might exhibit different characteristics compared to datasets involving individuals with normal hearing. However, this also presents an opportunity to investigate how our model generalizes to a population with specific hearing difficulties.

In order to have a comparison with another available EEG network our dataset was used to run the Very Large Augmented Auditory Inference (VLAAI) network, available in [Accou et al., 2023]. The network predicts an audio envelope from an EEG and calculates the correlation, which makes it a good option to compare the correlation results. The code is also publicly published in GitHub. This network is based on fully connected and convolutional layers.

The attended correlation results using the same dataset as in this project in the VLAAI pretrained model is 0.0165. After training it further with the new data the new correlation result is 0.068, The model trained from scratch also obtained similar results. This means that the proposed model outperforms the VLAAI network in this aspect. The better results are also obtained with a considerably smaller network as the VLAAI network is composed of over 1.7 million parameters while the proposed model contains only around 257 thousand parameters.

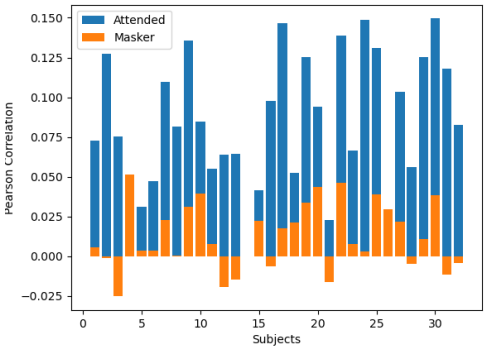


(a) Mean Attended Correlation per subject. We can see that the mean correlations, while above the null set, are not larger than the model trained with contrastive loss. Subject 14 was removed due to persistent artifacts.

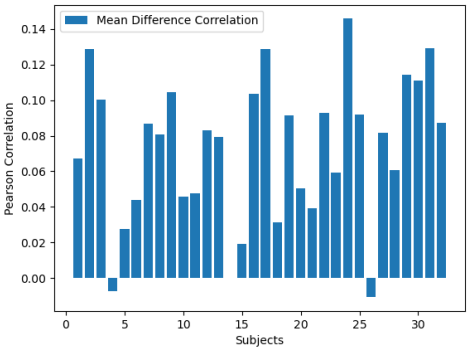


(b) Mean Masked Correlation per subject

Figure 6.4 Correlation mean per subject when training only on Pearson loss.



(a) Comparison of masked and attended correlation results. We can see that the average values of the correlations are much lower than if we used contrastive loss, and the maximum values are significantly reduced as well. The proposed model outperforms the model trained with only Pearson correlation as loss. Subject 14 was removed due to persistent artifacts.



(b) Difference between correlation of masked and attended envelopes with the predicted envelope.

Figure 6.5 Correlation comparisons Attended vs Masked when training only on Pearson loss.

7

Conclusion

In line with the proposed objectives, it was possible to incorporate contrastive learning into the context of auditory attention. The findings suggest that using contrastive learning can be advantageous for classifying auditory attention, with the potential to surpass existing models. It was proven that the model performs better in terms of accuracy when contrastive loss was applied as opposed to working with the Pearson correlation alone. Specifically, an accuracy of 68% and an attended correlation of 0.105 were obtained beating the proposed baseline.It is believed that further improvements in accuracy and correlation could be achieved through a more extensive hyperparameter search (which should include a large number of possible kernels as these were observed to be the most influential). Additionally, incorporating new layers with different connections could enhance the model's performance. The model presented here is relatively simple indicating significant potential for structural modifications and increased complexity. It is believed that a good reconstruction was obtained in terms of correlation.

Further Work

This is a good introduction to constrastive learning in the field of auditory attention but the authors of this project believe there is still space for improvement and further research. Some suggestions of possible tasks are as follows.

The model presented here uses only a limited variety of layers combined in a simple way. As mentioned, this is partly because of the shortage of data which makes it easy for a complex model to overfit. Nevertheless, further work should include a more thorough investigation on different layers, specifically what this new layers could bring to the model and how would this benefit the type of data being used. We also believe that there exist a different combination of either the current network or a more complex one that can boost the accuracy even more. Apart from this, a more expansive hyperparameter search could also yield better accuracy results.

During the project a very important decision for the model was the similarity metric. This metric is what indicates to the model how the training is going which means that if there was a similarity metric more suited to this type of data the model would

naturally achieve better results. This is why it is proposed to further investigate different similarity metrics in order to achieve a better loss function. This means not only to investigate and try different metrics but also combine them in such a way that better describes the necessity of the network.

The difference between subject independent and subject specific was explained earlier in the document, the results later presented were all obtained subject specific. It is proposed to test the model on subject independent scenarios. This includes testing the current model and modifying it, for example, the subject layer should be modified in order to accept unknown subjects and respond accordingly. This possible future work should also attempt to provide improvements to obtain better results in regards to the current model and to other existing subject independent models.

8

Appendix

8.1 Detailed Results

The results for each individual subject for both baseline and the proposed model can be found on Table 8.1.

		Subject															
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**15**	**16**	**17**
Baseline	**Accuracy**	0.663	0.738	0.503	0.540	0.535	0.647	0.663	0.647	0.690	0.631	0.567	0.583	0.658	0.610	0.610	0.674
	Attended Correlation	0.079	0.124	0.027	0.067	0.045	0.107	0.114	0.077	0.123	0.099	0.056	0.031	0.065	0.058	0.041	0.115
	Masked Correlation	-0.008	-0.001	0.021	0.042	0.042	0.016	0.010	-0.030	-0.005	0.014	0.033	-0.009	-0.020	-0.009	-0.008	0.012
Model	**Accuracy**	0.573	0.646	0.635	0.552	0.562	0.698	0.760	0.667	0.844	0.760	0.625	0.656	0.719	0.594	0.521	0.708
	Attended Correlation	0.053	0.117	0.068	0.081	0.031	0.123	0.145	0.078	0.179	0.098	0.059	0.067	0.073	0.047	0.042	0.124
	Masked Correlation	-0.004	0.004	-0.018	0.041	-0.000	0.000	0.012	0.002	0.030	0.031	0.014	-0.024	0.008	0.027	0.001	0.010
		18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**Mean**
Baseline	**Accuracy**	0.529	0.647	0.668	0.573	0.552	0.608	0.706	0.620	0.551	0.759	0.588	0.674	0.717	0.668	0.604	0.626
	Attended Correlation	0.048	0.091	0.108	0.042	0.100	0.079	0.162	0.083	0.054	0.128	0.067	0.153	0.123	0.086	0.062	0.084
	Masked Correlation	0.022	-0.012	0.007	-0.005	0.053	0.015	0.003	0.023	0.038	-0.007	0.002	0.022	-0.006	-0.029	0.000	0.007
Model	**Accuracy**	0.490	0.750	0.740	0.583	0.496	0.920	0.854	0.635	0.469	0.833	0.635	0.885	0.875	0.927	0.656	0.686
	Attended Correlation	0.041	0.124	0.126	0.042	0.084	0.183	0.173	0.106	0.027	0.194	0.075	0.258	0.172	0.186	0.095	0.105
	Masked Correlation	0.035	0.035	0.047	-0.009	0.049	-0.002	0.002	0.05	0.038	0.018	-0.018	0.002	0.035	-0.019	-0.014	0.017

Table 8.2 Complete results for each subject

Bibliography

Accou, B., J. Vanthornhout, and H. e. a. Hamme (2023). "Decoding of the speech envelope from eeg using the vlaai deep neural network". *Scientific Reports* **13**. DOI: https://doi.org/10.1038/s41598-022-27332-2.

Alickovic, E., T. Lunner, F. Gustafsson, and L. Ljung (2019). "A tutorial on auditory attention identification methods". *Front. Neurosci.* **13**. DOI: 10.3389/fnins.2019.00153. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6434370/.

Alickovic, E., E. H. N. Ng, L. Fiedler, S. Santurette, H. Innes-Brown, and C. Graversen (2021). "Effects of hearing aid noise reduction on early and late cortical representations of competing talkers in noise". *Frontiers in Neuroscience* **15**. ISSN: 1662-453X. DOI: 10.3389/fnins.2021.636060. URL: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2021.636060.

Ansel, J., E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala (2024). "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf.

Biesmans, W., N. Das, T. Francart, and A. Bertrand (2017). "Auditory-inspired speech envelope extraction methods for improved eeg-based auditory attention detection in a cocktail party scenario". *IEEE Trans Neural Syst Rehabil Eng.* **25**:5. DOI: 10.1109/TNSRE.2016.2571900.

Boselli, S. and G. Sridhar (n.d.). *Auditoryattentionclassification-contrastivelearning*. URL: https://github.com/sofiboselli/AuditoryAttentionClassification-ContrastiveLearning.

Crosse, M. J., G. M. Di Liberto, A. Bednar, and E. C. Lalor (2016a). "The multivariate temporal response function (mtrf) toolbox: a matlab toolbox for relating neural signals to continuous stimuli". *Frontiers in Human Neuroscience* **10**, p. 604.

Crosse, M. J., G. M. Di Liberto, A. Bednar, and E. C. Lalor (2016b). "The multivariate temporal response function (mtrf) toolbox: a matlab toolbox for relating neural signals to continuous stimuli". *Frontiers in Human Neuroscience* **10**, p. 604.

Crosse, M. J., N. J. Zuk, G. M. Di Liberto, A. R. Nidiffer, S. Molholm, and E. C. Lalor (2021). "Linear modeling of neurophysiological responses to speech and other continuous stimuli: methodological considerations for applied research". *Frontiers in Neuroscience* **15**. ISSN: 1662-453X. DOI: 10.3389/fnins.2021.705621. URL: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2021.705621.

Falcon, W. and The PyTorch Lightning team (2019). *PyTorch Lightning*. Version 1.4. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

Lawhern, V. J., A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance (2016). "Eegnet: A compact convolutional network for eeg-based brain-computer interfaces". *CoRR* **abs/1611.08024**. arXiv: 1611.08024. URL: http://arxiv.org/abs/1611.08024.

Liaw, R., E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica (2018). "Tune: a research platform for distributed model selection and training". *arXiv preprint arXiv:1807.05118*.

Marrone, N., C. R. Mason, and G. J. Kidd (2008). "Evaluating the benefit of hearing aids in solving the cocktail party problem". *Trends in Amplification* **12**:4. DOI: 10.1177/1084713808325880. URL: http://dx.doi.org/10.1088/1741-2552/aca220.

McFee, B., M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thomé, F. Robert-Stöter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Åsen, J. Lim, A. Malins, D. Hereñú, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A.

Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, and W. Pimenta (2024). *Librosa/librosa: 0.10.2*. Version 0.10.2. DOI: 10.5281/zenodo.4923181. URL: https://doi.org/10.5281/zenodo.4923181.

Puffay, C., B. Accou, L. Bollens, M. J. Monesi, J. Vanthornhout, H. V. Hamme, and T. Francart (2023). "Relating eeg to continuous speech using deep neural networks: a review". *Journal of Neural Engineering* **20**:4, p. 041003. DOI: 10.1088/1741-2552/ace73f. URL: https://dx.doi.org/10.1088/1741-2552/ace73f.

Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever (2021). *Learning transferable visual models from natural language supervision*. arXiv: 2103.00020 [cs.CV].

Tanveer, M. A., M. A. Skoglund, B. Bernhardsson, and E. Alickovic (2024). "Deep learning-based auditory attention decoding in listeners with hearing impairment". *Journal of Neural Engineering*.

Thornton, M., D. Mandic, and T. Reichenbach (2022). "Robust decoding of the speech envelope from eeg recordings through deep neural networks". *Journal of Neural Engineering* **19**:4, p. 046007. DOI: 10.1088/1741-2552/ac7976. URL: https://dx.doi.org/10.1088/1741-2552/ac7976.

Vandecappelle, S., L. Deckers, N. Das, A. H. Ansari, A. Bertrand, and T. Francart (2021). "Eeg-based detection of the locus of auditory attention with convolutional neural networks". *eLife* **10**. Ed. by B. G. Shinn-Cunningham, J. O'Sullivan, and A. Dimitrijevic, e56481. ISSN: 2050-084X. DOI: 10.7554/eLife.56481. URL: https://doi.org/10.7554/eLife.56481.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention is all you need". DOI: https://doi.org/10.48550/arXiv.1706.03762.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* **17**, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Zhai, X., B. Mustafa, A. Kolesnikov, and L. Beyer (2023). *Sigmoid loss for language image pre-training*. arXiv: 2303.15343 [cs.CV].

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2024

Document Number
TFRT-6235

Author(s)
Gautam Sridhar
Sofía Boselli

Supervisor
Emina Alickovic, Eriksholm Research Centre, Sweden
Martin Skoglund, Eriksholm Research Centre, Sweden
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Auditory Attention Classification with Contrastive Learning

Abstract

Auditory attention detection is crucial for understanding speech in noisy environments, a challenge known as the "cocktail party problem." This project investigates the use of electroencephalography (EEG) to identify which speaker a listener attends to. EEG's portability and real-time recording capabilities make it a promising tool for practical applications.

We propose a novel neural network model for auditory attention detection using EEG data. The model reconstructs the attended speech envelope while simultaneously classifying attended vs. unattended speech. It incorporates a contrastive learning loss function (SigLIP), which, to our knowledge, has not been previously applied to EEG-based auditory attention detection. The model architecture combines convolutional, fully connected, and attention layers.

Evaluated on an EEG dataset with 31 subjects, the model achieves a mean accuracy of 68% and a mean correlation of 0.105 between the reconstructed and attended envelopes. This surpasses the baseline performance of linear methods (63% accuracy, 0.084 correlation). These results suggest the potential of contrastive learning for improving auditory attention detection accuracy, warranting further investigation.

Keywords
System identification, PI controller, Electrical grid, Notch filter, Automatic control, Grey Box, Simulink

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-46

Recipient's notes

Security classification

http://www.control.lth.se/publications/