Lund University

Faculty of Engineering LTH
Department of Biomedical Engineering

Brain-to-Language Model: Decoding Semantic Speech Representations from EEG via Cross-Modal Alignment

Master's Thesis in Biomedical Engineering

Ellie O'Brien and Julia Engström

Supervisor: Christian Antfolk
Co-supervisor: Martin Skoglund and Emina Alickovic
Company: Eriksholm Research Centre

June 2026

Contents

1 Introduction **1**

1.1 Problem Statement 1

1.2 Aim 2

1.3 Outline 3

2 Background **5**

2.1 Auditory Processing 5

2.2 Semantic Representation 7

2.3 Electroencephalography 8

2.3.1 Event-Related Potentials 8

2.3.2 Temporal Response Function (TRF) 11

2.4 Machine Learning 13

2.4.1 Neural Networks 13

2.4.2 Convolutional Neural Networks (CNN) 13

2.4.3 Transformers 14

2.4.4 Inductive Bias, CNN vs. Transformer 15

2.4.5 Principal Component Analysis (PCA) 16

2.5 Deep Learning for Time-Series Data 16

2.5.1 1D Convolutional Neural Network 16

2.5.2 Dilated Convolutions and Receptive Fields 16

2.5.3 Gating Linear Unit (GLU) 17

2.6 Whisper 18

2.7 Multimodal alignment 19

2.7.1 Contrastive Learning 19

2.7.2 Retrieval Task 20

2.8 Related Work 21
2.8.1 Neuro2Semantic 21
2.8.2 Decoding Speech Perception from Non-invasive Brain Recordings 22
2.8.3 A Penny for Your Thoughts 22

3 Dataset **25**
3.1 Experimental Conditions 25
3.2 Data Acquisition 26

4 Methodology **27**
4.1 Implementation and Setup Details 28
4.2 Data Preprocessing 28
4.3 Creating Whisper Embeddings 28
4.4 Modelling TRFs 29
4.5 Model 1: Dilated CNN Encoder 32
4.5.1 Training 34
4.5.2 Model Architecture 35
4.5.3 Evaluation Metric 37
4.6 Model 2: A Hybrid CNN-Transformer Encoder 37
4.6.1 Data Preprocessing and Data Loading 39
4.6.2 Model Architecture 41
4.6.3 Evaluation Metric 43

5 Results and Discussion **45**
5.1 Performance of all TRFs 45
5.2 Evaluation and Comparison of Model Architectures 50
5.2.1 Model 1: Dilated CNN Encoder 50
5.2.2 Model 2 56
5.2.3 Model Comparison 62
5.3 Limitations 63
5.4 Future work 66

6 Conclusion **69**
References 71
Appendix i

Abstract

A key limitation of current hearing devices is the lack of user feedback to determine what speech content is actually being registered by the brain. To provide an objective measure of this cognitive processing for future brain-steered hearing devices, this study develops a brain-to-language model capable of decoding semantic-like information directly from non-invasive electroencephalography (EEG) signals. After modelling temporal response functions to confirm the correlation between Whisper embeddings and EEG signals, two distinct EEG encoders are trained via contrastive learning. Specifically, a dilated CNN architecture and a transformer model are utilised to map continuous neural data directly to the contextualised semantic embeddings extracted by Whisper. Since there is no observable ground truth for human semantic processing, this study is using Whisper's artificial embeddings to represent them. Despite this limitation, results demonstrate that meaningful semantic patterns can be extracted from EEG recordings, with both models performing above chance. The transformer architecture achieved the best results by successfully mapping neural signals to the uncompressed semantic space, outperforming the dilated CNN which required dimensionality reduction.

Popular Science Article

Extracting Speech and Meaning from Brain Signals

Hearing loss is a growing global challenge that demands smarter technology. This thesis investigates the use of artificial intelligence to match the patterns of brainwaves directly with the features of sound in order to decipher what a listener truly perceives and understands.

Hearing loss does not only make sounds fainter, it often distorts speech, making it incredibly exhausting for the brain to process and interpret the meaning of words. While modern hearing aids excel at amplifying sound, adjusting them still relies heavily on the user's subjective feedback and personal guesswork. This creates a critical need for smart, brain-steered hearing aids that can objectively measure how well a person actually understands speech.

In this work, we address this problem by designing two distinct models with the purpose of extracting patterns in brain signals and comparing these with characteristics in speech that have been extracted using a known transcription model. These comparisons are then used to train the models to find stronger similarities between the brain signal and the audio signal. If a high similarity is achieved, this suggests that the brain has successfully perceived the true meaning of the speech.

However, reading the human brain is no easy task. Brainwaves recorded through electroencephalography (EEG) are notoriously noisy and heavily contaminated by muscle movement, eye blinks and external interference. Because of this large amount of background noise, the absolute accuracy of matching the brain signals to the speech data might at first glance appear low. But when compared to random guessing, our AI models performed many

times better than chance. This clear margin suggests that the models are not just making lucky guesses, but have actually managed to extract language patterns from the messy brain data.

Looking ahead, this suggests that an objective measure of comprehension can be extracted from brain waves. However, an important question remains: the lack of a ground truth, meaning that we cannot be entirely sure of what language or semantic information the AI speech model actually captures. Yet, by proving that these signals can be aligned far better than chance, this work opens the door for future hearing aids to monitor the brain in real time for automatic adjustments of the hearing aid settings.

Author Contribution

Both authors contributed equally to the overall workload of this project. Initially, E. O'Brien focused on the software development, implementing the speech model, while J. Engström focused on the literature study. Subsequently, the technical workload was divided equally, with both authors responsible for coding one additional model each. The writing and editing of the final report were distributed equally between both authors.

Use of AI

The authors utilised generative AI tools to assist in three main areas of this project: summarising scientific reports to gain a rapid overview of relevant research, providing conceptual guidance and design ideas during code development, and improving grammatical clarity during the drafting of the report. All AI suggestions were evaluated critically, verified by the authors, and adjusted to ensure academic and technical accuracy.

Acknowledgements

We would like to extend our deepest gratitude to everyone who has supported and guided us throughout the journey of our thesis work.

First and foremost, we offer our sincere thanks to Eriksholm Research Centre for providing us with the opportunity to work on this incredibly interesting project. A special thanks to our supervisors Martin Skoglund and Emina Alickovic, whose guidance and support helped this project succeed.

We are especially grateful to Johanna Wilroth for her generous time and invaluable assistance in navigating the code and dataset challenges we encountered along the way. Furthermore, we would like to thank Oskar Keding for introducing us to the dataset and helping us get started with the code.

We also want to thank our friends for the breaks and fun times during all our lunches in the E-house.

Finally, we would like to express our appreciation to our academic supervisor, Christian Antfolk at LTH, for his constructive feedback on our report and for providing us with necessary resources.

1

Introduction

1.1 Problem Statement

Hearing impairment is a rapidly growing global health issue that, according to the World Health Organization (2021), affects 1.5 billion people worldwide, with 430 million individuals currently requiring rehabilitation services. Beyond the immediate auditory deficit, hearing loss significantly hinders verbal communication and language development, particularly in children, and frequently leads to social isolation, cognitive decline, and compromised mental health in adults. Consequently, advancing the capabilities of modern hearing assistive technologies is of utmost importance to mitigate the personal impacts of this condition.

Traditional hearing aids, while effective at amplifying sound, lack objective feedback mechanisms to verify whether the user has accurately perceived the speech, which becomes a major issue in complex environments where background noise drowns out the target speaker. To overcome this limitation, recent research has turned toward neural speech decoding, which involves extracting and reconstructing linguistic or semantic representations directly from non-invasive neural signals like electroencephalography (EEG). By understanding how the human brain processes and prioritises ongoing speech,

it opens up the possibility of designing a brain-steered hearing aid. In such a cognitive hearing aid framework, neural decoding would act as an objective measure of speech comprehension. Rather than relying on subjective user feedback, the system aims to continuously monitor the user's neural tracking of the audio stream to provide an objective indicator of how the target speech is being processed by the brain. This real-time objective metric can then be used as a feedback loop to automatically adjust and optimise the hearing aid's processing parameters.

While the profound potential of integrating neural speech decoding into assistive devices is clear, mapping continuous, high-noise EEG data to complex language representations remains a challenge. Various machine learning approaches have been proposed to model this relationship. A comprehensive review of these previous methodologies and their connection to our work is presented in Section 2 under Related Work.

1.2 Aim

The primary objective of this study is to develop a brain-to-language model, that is capable of decoding semantic speech features from neural signals, thereby linking neural activity directly to language representation. By using self-supervised speech models to extract contextualized representations from audio data, we aim to train deep learning models that can map neural features onto these semantic speech features for contrastive learning.

This work is conducted to gain a deeper understanding of the auditory processing mechanisms that the brain employs during continuous speech perception. By linking neural activity directly to continuous language representations, this research aims to provide a foundation for the future development of next-generation hearing aids.

To address these aims, this study will include the following objectives:

- Compute Temporal Response Functions (TRFs) to investigate how well the Whisper speech model can reconstruct neural activity from audio

data.

- Extract semantic embeddings from audio and EEG utilizing Whisper and designing an EEG Encoder, respectively.
- Train the models using contrastive learning.
- Evaluate the model retrieval performances and generalization capabilities.

1.3 Outline

The remainder of this report is structured as follows. First a necessary theoretical background is provided, introducing key concepts and relevant research within auditory processing, machine learning, and contrastive learning. Secondly, the dataset is introduced followed by the methodology, providing a detailed description of the proposed encoder model architectures. Finally, the experimental results and the model performances will be presented, followed by a comprehensive discussion and suggestions for future work.

2

Background

2.1 Auditory Processing

One of the most crucial mechanisms allowing humans to interact with their surroundings is the perceptual processing of auditory information (Drosos et al., 2024). Auditory processing is the process in which auditory information is interpreted and includes mechanisms that allow speech to be discriminated from non-speech, as well as the decoding of temporal and spectral features (Drosos et al., 2024; Jeddi et al., 2025).

Speech signals contain complex acoustic information that must be decoded by the auditory system in order to be understood. During early auditory processing, the cochlea decomposes incoming sound into its frequency and time components. This process allows listeners to extract both spectral and temporal cues that are essential for speech perception (Jeddi et al., 2025).

Two fundamental dimensions of speech processing are therefore spectral and temporal analysis. Spectral processing concerns the interpretation of frequency-related data within sound signals. Central to this is spectral resolution, the auditory system's precision in differentiating between various frequencies. As speech sound vary in their spectral characteristics, this becomes a crucial part in accurate speech perception (Jeddi et al., 2025).

Temporal processing refers to the ability of the auditory system to track dynamic fluctuations in acoustic signals as they unfold over time. A key component is temporal resolution, which describes how accurately a listener can perceive rapid shifts in amplitude or spectral content. This capability is particularly important for speech comprehension, as spoken language consists of rapidly changing acoustic patterns (Jeddi et al., 2025).

Both temporal and spectral processing contribute to the extraction of meaningful information in speech signals. Their importance is also reflected in computational approaches to speech processing, such as automatic speech recognition systems, which rely heavily on spectral features to model and interpret speech (Jeddi et al., 2025).

Auditory perception is not determined solely by the acoustic properties of sound but is also influenced by higher-level cognitive processes such as attention and memory. This interaction, in which perception is dynamically shaped by higher-order brain regions, is commonly referred to as top-down processing (Kraus and Banai, 2007).

At the neural level, the auditory cortex is organised in a tonotopic manner, meaning that different cortical regions respond selectively to specific sound frequencies. Importantly, the auditory system is also capable of neural plasticity, meaning that its functional organisation can change as a result of experience or perceptual learning (Kraus and Banai, 2007).

Neural plasticity also plays an important role in the context of hearing loss. When auditory input is reduced or degraded, the brain may adapt by reorganising its processing strategies in order to compensate for the loss of sensory information. These adaptations can influence how sounds are perceived and may contribute to differences in speech perception performance among individuals (Kraus and Banai, 2007).

2.2 Semantic Representation

Semantic representations are more than just definitions; they are mental states that serve two main purposes (Frisby et al., 2023). First, they help us see conceptual similarities, allowing us to connect two things even if they look completely different. For example, Frisby et al. (2023) mentions how we understand that both an ostrich and a hummingbird are birds, despite their huge differences in size and appearance. Second, semantics allow us to "fill in the blanks" by assuming traits about something even when they are not immediately visible. Another example from the authors is looking at a photo of a stationary parrot and knowing it has the ability to fly.

Furthermore, Frisby et al. (2023) describes three main ways that this conceptual knowledge is encoded. The first is category-based theories, where knowledge is sorted into separate bins, like the concept of a "boat" or a "tree." The second is feature-based theories, where concepts are built from specific properties. In this view, two things are similar if they share many features—such as an ostrich and a hummingbird sharing wings and feathers. The third approach, which is central to this thesis, focuses on latent space theories. Similar to common machine learning methods, these models map concepts into a shared space. Unlike specific features, the individual dimensions of this space do not have an obvious, readable meaning. Instead, similarity is determined by distance: if two points are close to each other in this latent space, it means the underlying concepts have a similar meaning.

In these models, inference relies on the direction and distance between vectors in the space rather than explicit labels. This principle also enables representations derived from different modalities to be aligned within the same latent space. When separate encoders are trained to map inputs such as images, audio, or neural signals into a shared embedding space, semantically related inputs can be located near each other regardless of their modality. This process is commonly referred to as cross-modal alignment (Frisby et al., 2023).

2.3 Electroencephalography

Electroencephalography (EEG) is a non-invasive measuring technique that captures electrical potential changes generated by brain activity. Electrodes placed on the scalp detect and amplify the temporal summation of postsynaptic potentials from synchronously firing neurons in response to stimuli. In the context of auditory perception, changes in cortical activity elicited by the onset or offset of auditory stimuli provide quantifiable measures of how the higher-order auditory cortex responds to and processes different sounds (Zhou et al., 2025). The recorded signals can then be further processed and analysed using various classification methods (Chaddad et al., 2023; Nayak and Anilkumar, 2025; Zhou et al., 2025). Oscillatory EEG waves can be described based on several key features including frequency, amplitude, shape of the waveform, and spatial origin (Nayak and Anilkumar, 2025).

As EEG provides an accessible measurement of the brain activity, it also has its limitations. Being an non-invasive technique with electrodes placed on the scalp, it introduces the risk of multiple physiological artifacts, such as eye movement, and muscle movement, yielding a low signal-to-noise (SNR) ratio. Furthermore, individual factors, the non-linearity of the signal and its unique characteristics contribute to the difficulty of acquiring high resolution signals (Chaddad et al., 2023).

2.3.1 Event-Related Potentials

Event related potentials (ERPs) are small voltage fluctuations evoked by stimuli events that reflect the synchronised postsynaptic activity of large neuronal populations engaged in stimulus processing (Sur and Sinha, 2009). ERPs are comparably smaller electrical potentials that are embedded in the EEG recordings, so to achieve a sufficient signal-to-noise ratio (SNR) and get the ERP to correspond to a specific stimulus, the neural responses are time-locked to an event and averaged across repeated stimulus presentations (Rugg, 2009; McWeeny and Norton, 2020). ERP components are typically characterised

by their voltage deflection and latency, where for instance P300 describes the positively deflected peak after 300 ms, and N100 describes a negative peak after approximately 100 ms. Furthermore, the latency of the the peaks provides insight into the temporal dynamics of cognitive processing stages (McWeeny and Norton, 2020).

P1

Early on in the cortical auditory response, irrelevant and redundant auditory information is being filtered out through a process called sensory gating, allowing selective attention to important stimuli. This generates a positive wave approximately 50 ms after the stimulus was presented, called the P1 or P50 component (Sur and Sinha, 2009; McWeeny and Norton, 2020).

N1

Another auditory sensory component is the N1 or N100 wave. This negatively deflected wave with its maximum in the frontocentral parts of the brain, corresponds to changes in the auditory stimulation, such as onset and offset of sounds (Sur and Sinha, 2009; McWeeny and Norton, 2020).

P2

Often visible after the N1 peak is the P2 or P200 component, creating a N1-P2 auditory component complex. An example of the N1-P2 complex also including P1, is displayed in Figure 2.1. The P2 component often arises in frontocentral part of the brain around 100-250 ms after stimulus onset and is thought to be related to attention (Paiva et al., 2016).

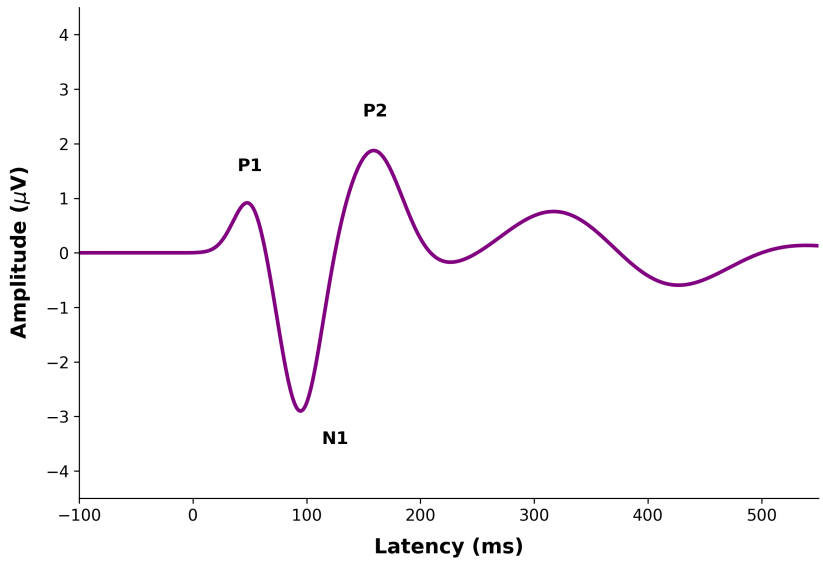


Figure 2.1. Example of an ERP illustrating the P1-N1-P2 complex.

N2

The N2 or N200 component is associated with cognitive processes such as selective attention and perception, typically elicited around 180 and 325 ms after the onset of an unexpected change in a repeated stimulus. This frontocentrally distributed peak comprises several distinct subcomponents (Patel and Azzam, 2005).

One such subcomponents directly tied to auditory processing is the mismatch negativity (MMN, or N2a) component (Patel and Azzam, 2005). The MMN reflects the difference between these deviations in sound and established response pattern. This is particularly relevant for studying the brain's capacity for auditory feature discrimination and its ability to track the probability of incoming stimuli (Garrido et al., 2009; McWeeny and Norton, 2020). Importantly, the MMN is elicited regardless of whether the subject is actively paying attention to the stimuli. In contrast, the N2b subcomponent is a negative peak not limited to auditory stimuli and is elicited only when the stimuli is selectively attended. Furthermore, while the MMN reacts to acoustic changes, the N2b has been shown to be closely linked to semantic processing

(Patel and Azzam, 2005).

P3

The P3 also known as the P300 peak is a response with connections to higher cognitive functions, such as attention, memory, and top-down processing. The P3 peak reflects an unexpected change in stimulus and is often most visible in frontal, central, and parietal parts of the brain (Polich, 2007).

N4

When an unexpected or semantically incongruent word is presented, the ERP waveform exhibits an increased negative deflection approximately 400 ms after stimulus onset. This component, known as the N4 or N400, is associated with reflecting the processing of semantic incongruity. Consequently, the N4 serves as a valuable measure for investigating semantic relationships and examining interindividual differences in semantic processing (McWeeny and Norton, 2020).

2.3.2 Temporal Response Function (TRF)

Temporal Response Functions (TRFs) are widely used in signal processing and cognitive neuroscience to study the relationship between continuous stimuli and neural activity. They are commonly applied when analysing natural input, such as speech, that unfolds over time. Instead of focusing on isolated events, TRFs model how time-varying stimulus features relate to ongoing neural responses. These TRFs can be analysed in a similar way to the ERPs mentioned above.

A TRF can be understood as the brain's impulse response within a linear framework. This means that the neural signal is modelled as the result of a convolution between the stimulus and a filter function. The filter corresponds to the TRF. In practical terms, the TRF describes how the brain responds over time to a brief change in the stimulus. It therefore captures the temporal dynamics between stimulus events and neural activity (Brodbeck et al., 2023).

The TRF can be estimated using the boosting algorithm provided by the

Eelbrain Python toolbox Brodbeck et al. (2023). Unlike the ensemble boosting methods typically associated with machine learning, this algorithm iteratively updates weights to fit a sparse linear model. The training process initialises all TRF weights to zero. To evaluate performance and prevent overfitting, the dataset is partitioned using K-fold cross-validation. Conceptually, the model can be formulated as:

$$y_t = \sum_{\tau=\tau_{min}}^{\tau_{max}} h_\tau x_{t-\tau} \tag{3.1}$$

where y_t is EEG at a certain time t, $x_{t-\tau}$ is the stimulus at an earlier time, and h_τ is the TRF-weight at the lag τ. Equation 3.1 shows the idea of TRF and how it is modelled. It is built on a linear model of the neural signal y_t and is explained by representations of stimulus x_t at different time delays τ. The TRF weights h_τ describes how much every time delay affects the signal.

The estimated weights form the temporal response function, which describes the brain's impulse response to the stimulus features. Each weight corresponds to a specific predictor-lag combination and reflects the contribution of that stimulus feature to the neural signal at a given delay. Positive weights indicate that increases in the predictor are associated with increases in neural activity at the corresponding lag, while negative weights indicate an inverse relationship (Brodbeck et al., 2023).

When boosting is used to estimate the TRF, the weights are updated iteratively rather than solved analytically. At each step, the algorithm identifies the predictor-lag combination that produces the largest reduction in prediction error and adjusts the corresponding weight accordingly. This process continues until further updates no longer improve model performance on validation data. Because only informative predictor-lag combinations receive weights that are not zero, boosting produces sparse TRF solutions in which irrelevant predictors or lags effectively are excluded from the model (Brodbeck et al., 2023).

2.4 Machine Learning

Machine Learning enables computers to perform tasks without being explicitly programmed. By using algorithms to identify patterns in large datasets, the system can be trained to create models capable of making accurate predictions and classifications (França et al., 2021).

2.4.1 Neural Networks

An Artificial Neural Network (ANN) is a machine learning model inspired by signal propagation in a biological neural network. It comprises an input layer, one or more hidden layers, and an output layer, all made up of interconnected nodes with learnable weights. The training process involves iterative adjustments of the connection weights to minimise the error between the predicted and actual outputs (Han et al., 2018; Walczak and Cerpa, 2003).

2.4.2 Convolutional Neural Networks (CNN)

In machine learning, a common technique used for pattern recognition and classification is deep learning. Deep learning models consist of multiple hidden layers in addition to the input and output layers, which enables important features to be extracted from hierarchical architectures. However, the performance of the algorithm benefits from a large amount of data (Sarker, 2021). Convolutional Neural Networks (CNNs) are one type of deep learning model commonly used for image classification and recognition and have demonstrated strong feature learning abilities.

A CNN architecture typically consists of a series of alternating convolutional and pooling layers, which function as a hierarchical feature extractor, followed by one or more fully connected layers that perform the final classification or regression task. The convolutional layer extracts features from the input data by implementing filters, also called kernels. These kernels consist of weights and slide over the input data, where the scalar product between the kernel and the input results in a feature map. To introduce non-linearity and allow the

network to learn complex matters, a non-linear activation function is applied directly to this feature map (Taye, 2023).

The stride of the convolution determines the overlap when the kernel slides across the input. Altering the overlap leads to changes in the feature map's dimensionality, as an increasing stride reduces overlap and consequently decreases the output dimensions. The size of the feature map can be further modified by adding padding. Padding ensures that the spatial information at the edges of the input is preserved, thereby increasing the size of the resulting feature map. In the pooling layer, the aim is to decrease the dimensionality of the feature map while still maintaining the important data. The dimensionality reduction is performed through downsampling by sliding a filter over the data, where the most commonly used method is called max-pooling (Taye, 2023; O'Shea and Nash, 2015). Lastly, the fully-connected layers are responsible for classification and recognition (Zhang et al., 2018).

2.4.3 Transformers

Unlike traditional sequential architectures that process data step by step, the Transformer is built entirely on self-attention. This design allows the model to process all parts of a sequence simultaneously, enabling significantly faster parallel processing during training (Vaswani et al., 2017).

Since the Transformer lacks an inherent sense of time or order due to its non-sequential nature, it utilises positional encodings. These are added to the input embeddings to provide the model with essential information regarding the position of each token in the sequence (Vaswani et al., 2017).

The most fundamental part of the architecture is the self-attention mechanism. This allows the model to weigh the importance of different tokens in a sequence when processing a specific element, effectively capturing the context. This is achieved by mapping the input into three distinct matrices: Queries (Q), Keys (K), and Values (V). The relationship between the words is calculated using the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{4.2}$$

In this equation, the dot product of Q and K determines the alignment between tokens. To prevent the gradients from vanishing during training at larger dimensions, the product is scaled with $1/\sqrt{d_k}$, where d_k is the dimension of the keys. The softmax function then normalises these scores, before they are used to weigh the values in V (Vaswani et al., 2017).

To further enhance the model's contextual depth, the Transformer employs multi-head attention. Rather than performing a single attention function, the model runs multiple heads in parallel, allowing a greater variety of relationships to be captured. The final architecture consists of a multilayer encoder and decoder that integrates these multi-head attention blocks and feed-forward networks to refine the data representations (Vaswani et al., 2017).

2.4.4 Inductive Bias, CNN vs. Transformer

When comparing Convolutional Neural Networks (CNNs) and Transformers, a fundamental distinction lies in their inductive biases. CNNs operate with a strong inductive bias, relying on convolution-induced constraints that assume local connectivity within the data. This structural assumption makes CNNs efficient at extracting local patterns but can restrict their ability to integrate information across larger distances (Deininger et al., 2022).

In contrast, Transformers possess a weaker inductive bias. Because they are free of convolution-induced biases, they allow for more flexible feature detection. This architectural freedom enables the attention mechanisms within a Transformer to learn global features and identify complex relations across the entire data sequence (Deininger et al., 2022).

2.4.5 Principal Component Analysis (PCA)

The high dimensionality of the dataset introduces significant computational demands and increases the risk of model overfitting. To avoid this, principal component analysis (PCA) is introduced, which is used as a dimension reduction technique. PCA makes it possible to reduce dimensions while preserving as much statistical information (variance) as possible (Jolliffe and Cadima, 2016).

2.5 Deep Learning for Time-Series Data

2.5.1 1D Convolutional Neural Network

While two-dimensional CNNs are designed for spatial data like images, one-dimensional (1D) CNNs are specifically tailored for continuous time-series data. In a 1D CNN, the input is typically represented as a sequence, and the convolutional kernel slides along this single time axis.

As the kernel moves, it operates on short, consecutive segments of the input to extract local patterns. Stacking multiple convolutional layers makes hierarchical feature learning possible. This means that the early layers capture small, local temporal features, while the deeper layers integrate these into broader, more abstract representations of the signal (Kiranyaz et al., 2021).

2.5.2 Dilated Convolutions and Receptive Fields

Standard convolutional networks often capture longer temporal patterns by using pooling or subsampling layers to compress the data. However, this progressively reduces the resolution of the output. This loss of temporal resolution is a disadvantage for prediction tasks that require a precise, moment-to-moment mapping of the input signal (Yu and Koltun, 2016).

To aggregate a wider context without losing resolution, networks can use dilated convolutions. A dilated convolution introduces gaps between the values in a filter. This allows the same filter to cover a larger area of the

input sequence. By systematically increasing the dilation factor in successive layers, the network's receptive field expands exponentially without shrinking the original input data (Yu and Koltun, 2016).

Crucially, expanding the receptive field this way does not require additional parameters. Since the filter simply spreads out rather than increasing in size, the number of learnable parameters associated with each layer remains identical to a standard convolution. Consequently, the receptive field can grow exponentially with network depth, while the number of parameters grows only linearly (Yu and Koltun, 2016).

2.5.3 Gating Linear Unit (GLU)

The Gated Linear Unit (GLU) is a data-dependant multiplicative gating mechanism designed to control information flow within deep convolutional networks. Given a 1D convolution that doubles the channel dimension, producing a tensor that is split up into two equal halves A and B, the GLU is defined as:

$$GLU(A, B) = A \odot \sigma(B) \tag{5.3}$$

In this simplified equation from Dauphin et al. (2017), B passes through a sigmoid function (σ), transforming it into a "gate" with values between 0 and 1. This gate then modulates A through element-wise multiplication ($\odot$) as seen in Figure 2.2, deciding which features are preserved and which are suppressed. This mechanism is effective for processing low-SNR signals such as EEG because it allows the model to dynamically suppress noise while maintaining the neural patterns.

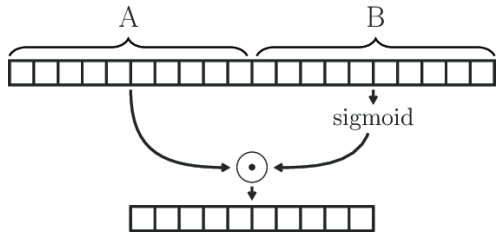


Figure 2.2. Gated linear unit visualised by Elbayad (2020).

This specific pattern of dilated residual blocks and GLU gating is utilised in Défossez et al. (2023) to decode perceived speech from non-invasive neural recordings, where the GLU acts as a primary tool for learned noise suppression.

2.6 Whisper

To extract rich speech representations from audio stimuli, this study utilises Whisper, a multitask, Transformer-based encoder-decoder model developed by Radford et al. (2022). Whisper is trained on 680,000 hours of weakly supervised, large-scale audio data, enabling it to transcribe speech across 99 different languages (Anderson et al., 2023).

The audio processing pipeline begins by resampling the raw audio to 16 kHz and converting into an 80-channel log-Mel spectrogram. This conversion is performed using 25 ms analysis windows with a 10 ms time step, providing a time-frequency representation of the audio files that closely resembles human auditory perception (McLoughlin et al., 2026). Prior to network processing, these spectrogram features are uniformly rescaled to a [-1, 1] range, which centers the input distribution around zero and optimises the gradient flow during training.

Before entering the Transformer encoder, the normalised features are preprocessed by two convolutional layers, and sinusoidal position embeddings are added to provide crucial time-point references. The features are subsequently processed by multiple layers of Transformer encoder blocks. Within these

blocks, each layer utilises a self-attention mechanism combined with a multi-layer perceptron (MLP) to refine and contextualise the acoustic features. This sequence of operations ultimately produces time-aligned linguistic output vectors (Radford et al., 2022; Anderson et al., 2023).

While the full Whisper architecture contains an autoregressive Transformer decoder designed to translate these vectors into text tokens (Radford et al., 2022), this study focuses exclusively on the contextualised embeddings generated by the Transformer encoder.

2.7 Multimodal alignment

2.7.1 Contrastive Learning

In this work, we use a contrastive objective to align EEG signals with its corresponding audio representation, similarly to (Défossez et al., 2023). Compared with regression, a contrastive loss directly encourages the model to assign higher similarity to matching brain–audio pairs than to mismatched pairs.

At its core, this means that the model learns by comparing. The fundamental objective of this approach is to map data points into a shared embedding space based on their relative similarities. During training, representations of "similar" (positive) samples are actively pulled close together, while representations of "dissimilar" (negative) samples are pushed further apart.

To achieve this mapping, the framework processes raw input data and transforms it into compressed, meaningful feature representations. Because the contrastive approach is highly flexible, it can be applied across different modalities, such as matching an audio signal to brain activity. In this thesis, the audio representations are extracted by the Whisper model, while the brain signals are processed by the proposed neural networks. By projecting these distinct types of data into a shared, lower-dimensional space, the model can effectively measure the distance between them and determine how conceptually related they are (Le-Khac et al., 2020).

2.7.2 Retrieval Task

The aim of this work's models is not to reproduce the exact acoustic structure of a sound from the EEG, it is to align the continuous neural signal with the semantic representations extracted by Whisper. This distinction matters for evaluation. Measuring word error rate via a text generator tests whether the model produces the right words in the right order which is a much stricter demand than measuring whether the model has understood the semantic content of the stimulus. A model can have a strong semantic representation of a word while still scoring badly on word error rate simply because the generator makes structural errors. On the other hand, a generator can produce fluent text that does not reflect the brain signal at all. Evaluating in the embedding space test semantic understanding directly, without these confusions.

Top-K Retrieval as a Semantic test

During evaluation, the model predicts an embedding $\hat{\mathbf{z}}$ from the continuous EEG signal. This predicted embedding is ranked against a large candidate pool of N unseen Whisper embeddings using cosine similarity:

$$\text{sim}(\hat{\mathbf{z}}, \mathbf{z}_i^a) = \frac{\hat{\mathbf{z}} \cdot \mathbf{z}_i^a}{\|\hat{\mathbf{z}}\| \, \|\mathbf{z}_i^a\|} \tag{7.4}$$

In this evaluation, the variable K defines the threshold of highest-ranked predictions considered for a successful match (e.g., $K = 1, 5,$ or 10). A successful top-K match occurs when the actual auditory stimulus presented to the participant falls within these top K candidate embeddings. Because this candidate pool is entirely unseen during training, accurate retrieval cannot be attributed to simple memorisation. Instead, it indicates that the predicted neural embedding successfully aligns with the contextualised speech representation. This interpretation is supported by Défossez et al. (2023), who demonstrated through a linear regression analysis of linguistic features that such decoded embeddings are driven primarily by word- and phrase-level semantic features rather than low-level acoustic properties. This validates

top-K retrieval accuracy as a measure of retrieval accuracy within the shared embedding space.

2.8 Related Work

2.8.1 Neuro2Semantic

The study by Shams et al. (2025) shares a similar objective to ours, except that they use intracranial EEG (iEEG) rather than non-invasive measurements. This significantly distinguishes their work from ours, as they do not face the same challenges regarding low SNR. Nevertheless, several of their methodological concepts remain highly relevant to our research. In their study, iEEG recordings are used to reconstruct the meaning of spoken language directly from neural signals. The aim is not only to detect words but to capture the underlying semantic content of speech. The Neuro2Semantic framework contains two main stages. First, neural signals are mapped to a semantic embedding space derived from pre-trained language models. Second, these embeddings are converted into natural language text. It is this first stage that is our main focus.

The neural-to-semantic mapping by Shams et al. (2025) is achieved using a Long Short-Term Memory (LSTM)-based encoder that transforms the neural signals into vector representations. These vectors are then trained to align with text embeddings generated by a pre-trained language model. The objective is to place the embeddings in the same semantic space.

To achieve this alignment, Shams et al. (2025) employ a contrastive learning strategy. During training, the objective function encourages the predicted neural embeddings to converge with their corresponding linguistic targets, while maximising the distance from unrelated embeddings within the shared latent space.

2.8.2 Decoding Speech Perception from Non-invasive Brain Recordings

A framework for decoding non-invasive continuous speech decoding was established by Défossez et al. (2023), who successfully mapped MEG and EEG recordings to complex speech representations. Their approach avoids direct text generation, instead they use contrastive learning objective to align neural embeddings with audio embeddings generated by a pre-trained, frozen wav2vec 2.0 encoder.

To handle the high variability between subjects and the continuous temporal dynamics, their "Brain Module" relies on two core architectural innovations. First, the raw data is passed through a 1x1 convolutional layer specific to the subject which acts as a personalised spatial filter. Since brain signals and sensor placements differ between participants, this layer projects the data into a shared embeddings space. By doing this, the deeper layers of the model can share weights and be trained on data from multiple people at the same time. Second, the temporal processing is done by using a stack of 1D convolutional blocks with dilated convolutions. This allows the network to have a larger receptive field, meaning it can capture the broader context of a 3-second speech window without losing the high temporal resolution of the raw signals. Furthermore, Gated Linear Units (GLU) are used to filter out background neural noise. The data is then projected into a dimension that matches the wav2vec 2.0 outputs.

Finally, to align the embeddings, the researchers had to handle the biological delay of speech perception. Because the brain processes sound slightly after it is actually heard, the input brain signals are artificially shifted 150 ms into the future. This provides a better baseline for the model to align the neural and audio data when using the contrastive loss function.

2.8.3 A Penny for Your Thoughts

A related approach is presented by Auster et al. (2025) in the paper *"A Penny for Your Thoughts"*, where the authors investigate whether neural networks can

be used to map EEG signals to audio representations in order to decode speech-related information from neural activity. In their work, embeddings derived from EEG signals and corresponding speech embeddings are projected into a shared embedding space, enabling the model to learn relationships between neural activity and speech representations. This study is particularly relevant as it explores strategies for extracting EEG embeddings and emphasises the importance of personalisation in improving decoding performance.

Starting from a baseline acquired from Meta (Meta Platforms Inc.), they introduce three modifications to the model architecture to improve model accuracy; subject-specific attention layers, personalised spatial attention, and a dual-path Recurrent Neural Network (RNN) including an attention layer.

The audio data was processed with a pre-trained wav2vec2 encoder, providing speech embeddings. To get corresponding neural embeddings from the EEG data, they used an encoder based on Meta's architecture. However, they modified several components including the adaptive spatial attention, subject-specific attention, stacked convolutions, dual-path Recurrent Neural Network (RNN) with attention, and final convolutions.

Instead of using a shared attention map for all subjects, this study introduced attention weights unique for each subject, allowing more personalised response. Furthermore they added a subject-specific layer that allowed patient specific patterns to be captured in the EEG.

To align and compare the created embeddings, the contrastive CLIP loss function was used when training the model.

3

Dataset

The dataset was originally introduced in Wilroth et al. (2026), where it was used to investigate neural tracking of selective auditory attention in audiovisual multi-speaker environments. In this work, the same dataset is used. The study includes 24 normal-hearing participants and was conducted using audiovisual multi-speaker listening scenarios. The experimental design contains sustained attention, attention switching, and conversational listening conditions.

The dataset includes 24 native Danish speakers (12 male), aged 23–51 years. All participants had normal hearing, confirmed by audiometric testing. The study was approved by the relevant ethics committees, and written informed consent was obtained from all participants prior to participation.

3.1 Experimental Conditions

The study consists of three experimental conditions. In the sustained attention condition, participants attended to one of two competing audiovisual speakers. In the switching attention condition, participants switched attention between two speakers twice within a trial. In the conversational condition, participants attended either to a two-speaker conversation or to a competing side speaker.

Each trial lasted 180 seconds and involved one condition.

3.2 Data Acquisition

EEG data were recorded using a 64-channel system with a modified Easycap and a Smarting Pro X amplifier. Twenty electrodes were replaced with cEEGrid electrodes, resulting in 44 scalp EEG channels and 20 around-the-ear channels. In this work, only the 44 scalp EEG channels are used. The sampling rate was 500 Hz, and electrode impedances were maintained below 20 kΩ throughout the recordings.

The speech stimuli consisted of pre-recorded audiovisual material presented via loudspeakers. The corresponding audio signals are also used in our work.

4

Methodology

This section outlines the methodology for developing models capable of decoding semantic information directly from brain activity. We present two distinct architectures based on cross-modal contrastive learning, both sharing the goal of extracting meaningful semantic representations from continuous EEG data. What separates the two models is mainly their approach to encoding the neural signals. Our method for generating audio embeddings using Whisper is inspired by (Radford et al., 2022; Anderson et al., 2023). The EEG-encoding strategy of our first model stems from Défossez et al. (2023) previous work in the field and our second model was designed with architectural inspiration from d'Ascoli et al. (2025). The dataset used to train and evaluate these models consists of 23 subjects, where each subject listened to nine audio trials lasting 178 seconds each, yielding approximately 10 hours of total audio data. The remainder of this section will describe the modelling of TRFs, the architecture of the two models mapping to Whisper's latent space, and finally present the contrastive training pipelines and the evaluation metrics. The hyperparameters used for both models are presented in the Appendix.

4.1 Implementation and Setup Details

All data processing, modelling, and evaluation pipelines were developed in Python using Visual Studio Code (VS Code). The neural network architectures were implemented using the PyTorch deep learning framework. While the TRF modelling and initial model development were conducted locally on standard hardware (8 GB RAM), this environment proved insufficient for the memory demands of high-dimensional embeddings and larger batch sizes. Consequently, the final training and evaluation phases were migrated to a high-performance computing environment. All model training was executed using an NVIDIA TITAN V graphics processing unit with 12 GB of VRAM. This hardware setup provided the necessary computational capacity to process the extensive continuous EEG data and maintain an effective batch size of 512 during the contrastive learning process.

4.2 Data Preprocessing

The EEG data were preprocessed in the same way as detailed in (Wilroth et al., 2026), with the exception that the only condition used in this project was the sustained attention (SustA).

4.3 Creating Whisper Embeddings

The audio data was processed using Librosa, an open-source Python library dedicated to audio signal analysis and feature extraction (McFee et al., 2015). To comply with the input specifications of the Whisper model, all audio was loaded at a sampling rate of 16,000 Hz. The total duration of the audio was calculated from the number of samples and the sampling rate.

The Whisper model used in this thesis was the medium variant, which has a hidden dimension of 1024. Testing additional model sizes is left as future work.

Since Whisper processes at most 30 seconds of audio at a time, embeddings

were extracted using consecutive, non-overlapping 30-second windows. Each chunk was passed to the Whisper processor, which converted the raw audio into a log-Mel spectrogram and returned it as a PyTorch tensor ready for the encoder. The last chunk might be shorter than 30 seconds, if that was the case, the processor handled this automatically by padding it to the expected length.

Gradients were disabled during this step, as the model was used for inference only which reduces memory usage and speeds up computation. Each chunk was passed through the encoder and the last hidden state was extracted, yielding a tensor shape of $[1500, 1024]$ after removing the batch dimension. Here, 1500 is the number of time steps produced by Whisper for a 30-second window and 1024 is the hidden dimension.

These 30-second segments were subsequently concatenated into a continuous NumPy array. Consequently, each 178-second auditory trial generated a corresponding Whisper embedding with a final tensor shape of $[8900, 1024]$.

Before returning the embeddings, the actual audio duration was used to trim any padding introduced by the final chunk. If the resulting frame rate differed from the target output sampling rate, the embeddings were resampled along the time axis using the Python module Resampy.

4.4 Modelling TRFs

As our initial thought was to use Whisper to extract the semantic representations from the audio data, we started by investigating Whisper's contribution to predicting the EEG signal from the audio data compared to just using the acoustic envelope, which is the slow changes in amplitude of the sound wave.

Feature extraction

The predictor variables used to model the TRFs were the acoustic envelopes extracted from the audio files as well as semantic features derived from the last hidden layer in the Whisper model.

The acoustic envelope is an effective feature for capturing the slow amplitude variations in speech over time (Issa et al., 2024). It was therefore extracted as a feature for the TRF analysis. The acoustic envelopes were extracted from the audio files as described in Wilroth et al. (2026). To extract the envelope, gammatone filtering was applied followed by bandpass filtering between 1-20 Hz, downsampling to 50 Hz, and normalisation. These predictors that we now call gammatone predictors, have a segment duration of 178 s, resulting in a total of 8900 samples.

During the generation of Whisper predictors, a Principal Component Analysis (PCA) was applied, retaining the first 10 principal components thereby reducing the dimensionality of the embeddings from 1024 to 10.

Both gammatone and Whisper predictors were saved as N-dimensional variables (NDVars), a N-dimensional data container that matches the time axis to the EEG, before estimating the TRFs with Eelbrains Boosting algorithm. A null model was created using the gammatone predictors together with Whisper predictors with mismatching subjects and audio.

TRFs were estimated using three different feature sets being: the null model, only gammatone predictors, and a combination of gammatone and Whisper features.

Boosting

For the TRF computation, the boosting-algorithm from the Eel-brain python toolkit by Brodbeck et al. (2023) was used. We implemented forward modeling using the boosting-based algorithm to derive sparse TRFs. In this approach, the model predicts the EEG signal based on stimulus features, effectively capturing the neural tracking of the input stimuli.

For each subject and model, the window of time-lags was set to span between -0.5 s before the stimulus to 1 s after, covering both early before the stimulus event and the later expected peaks. It was implemented using early stopping meaning that the algorithm stops updating it's weights when the error is no longer decreasing, done to prevent overfitting.

K-fold cross-validation is used by the boosting algorithm to ensure that the model is general and not overfitting. The data was split into nine segments for each subject, which corresponds to every trial. Of these nine segments, eight were used for training the boosting algorithm and the remaining segment was used as the independent testing set. This process was rotated nine times so that every data segment functioned as test data once, these partitions are shown in Figure 4.1. Every partition yielded a correlation coefficient, although the resulting coefficient is the average of the results from each respective partition.

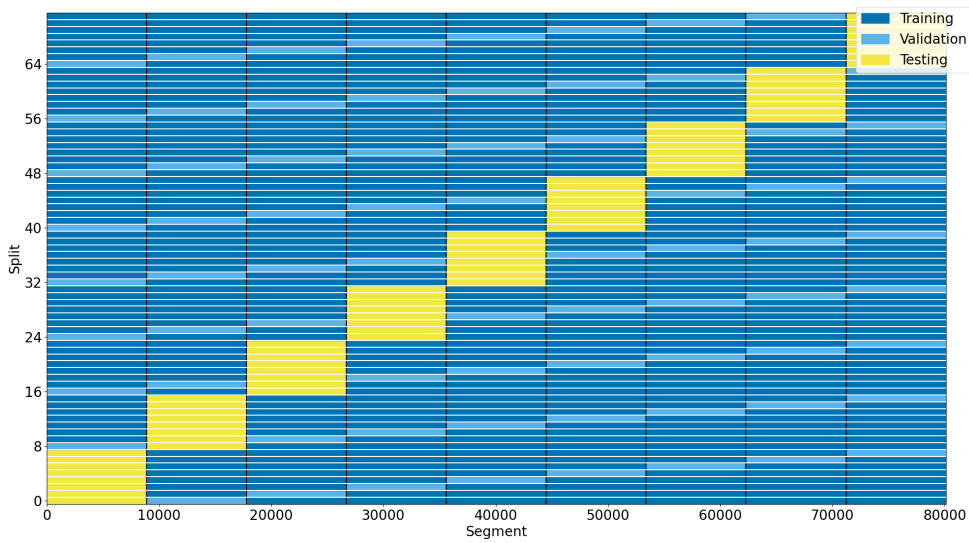


Figure 4.1. Data partitioning across 9 folds for iterative training, validation, and testing.

The change in weights (h) over time was visualised using Eelbrain's topoButterfly plot to illustrate the predicted neural responses derived from the acoustic envelope and Whisper embeddings. These visualisations can be analysed similarly to standard ERPs. The topoButterfly plots also provide a topographic map showing in which sensors the connection between audio and brain is the strongest.

Evaluation by Pearson Correlation

The predicted EEG was compared to the real EEG signals using Pearson

correlation. Pearson correlation compares the similarity between two objects, generating a score (Pearson's coefficient (r)) between -1 and +1, where a score near +1 indicates high similarity, and -1 suggests an inverse correlation. If the two data object are uncorrelated it results in a score close to zero (Berman, 2016). As there is one r-value for each sensor per subject and the correlation was calculated as a mean across all subjects, the resulting correlation is defined per sensor.

4.5 Model 1: Dilated CNN Encoder

To extract semantic representations from EEG data and project into Whisper's dimension space, a deep convolutional neural network (CNN) architecture inspired by Défossez et al. (2023) was implemented. While both models use similar dilated convolutional blocks, the proposed architecture introduces a few changes.The main difference is the target space. Instead of using wav2vec 2.0 based on acoustic features, this work targets the Whisper embedding space, which required adapting the network for the new dimensions. Furthermore, to handle differences between subjects, this model compares two initial projection strategies: the original subject-specific 1×1 spatial filters, and a shared 1×1 projection layer combined with a learnable subject embedding. Finally, instead of using spatial dropout on specific sensor regions, standard channel dropout is applied directly inside the residual blocks to reduce overfitting.

The model functions as a neural encoder that maps a temporal window of EEG signals to a predicted continuous semantic vector. The training objective is to align these predictions with their corresponding ground-truth Whisper latent representations within a shared embedding space.

The architecture of the proposed model is illustrated in Figure 4.2. The pipeline consists of three main parts: a subject-specific initial projection to handle inter-individual variability, a series of dilated convolutional blocks for temporal feature extraction, and a final projection layer that outputs the predicted semantic vector.

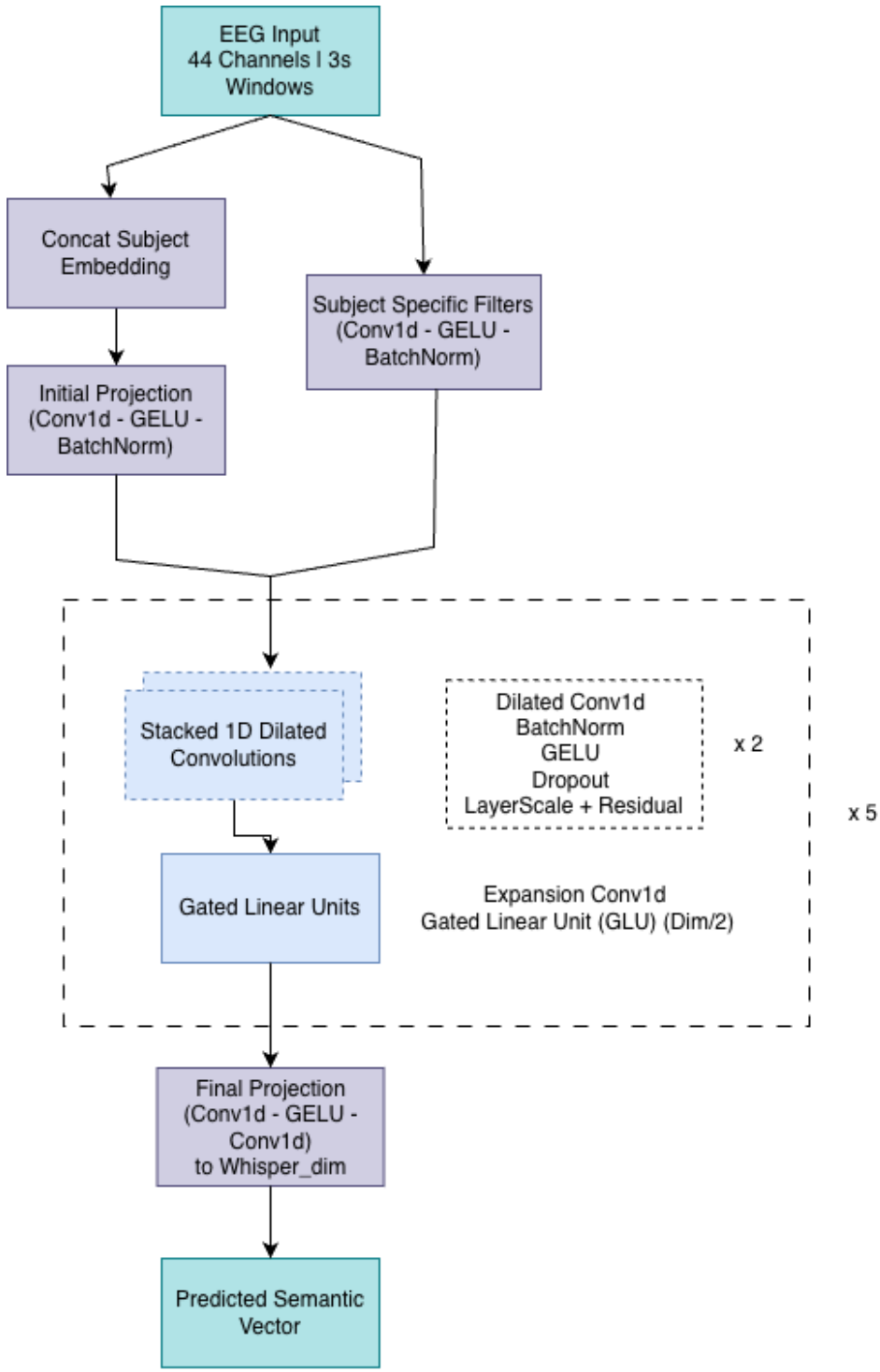


Figure 4.2. Model 1, overview of architecture.

4.5.1 Training

Batch Construction and Data Alignment

When working with time-series data and an architecture relying on aligning two modalities, EEG data with the audio-based Whisper embeddings, the handling of batches and data alignment becomes vital for the objective of the work. This alignment was preserved through training by using rigid index batching. To ensure temporal alignment, the pre-processed EEG recordings and their corresponding Whisper embeddings were segmented using a sliding window approach with identical temporal boundaries (window size = 3 seconds). To prevent semantic overlap and avoid false negatives in the contrastive objective, the step size was set equal to the window size (step size = 3 seconds).

During the data loading phase, the dataset yielded synchronised pairs of EEG windows and Whisper embeddings. When constructing a training batch, the dataloader stacked these pairs such that the input tensor and the target tensor maintained parallel indexing. This means that the i-th sample in the EEG batch corresponded to the i-th sample in the Whisper embedding batch.

When the EEG batch was processed by the models, it was passed through sequentially, keeping the same index as before it had passed through the network. Finally, when calculating the loss, the criterion used this parallel structure: for any prediction at index i, the loss function treated the target embedding at index i as the positive match. Since the windows were non-overlapping, the loss function can treat all other targets within the same batch as negative samples, ensuring the network is trained on the true temporal and semantic ground truth.

Loss function

To optimise the alignment between the neural signals and the audio representations, this work implements a symmetric, sequence-level contrastive loss adapted from the CLIP architecture (Radford et al., 2021). Given a predicted EEG-derived embedding and a candidate Whisper embedding, the model first

L2-normalises both representations and computes their mean cosine similarity across the temporal dimension. A bidirectional cross-entropy objective is evaluated to calculate the loss for both brain-to-audio and audio-to-brain retrieval. These two losses are averaged to ensure a strict and robust alignment of the two modalities within the shared latent space.

Data Splitting

The split was based on the trials, for each subject seven trials are set in the training dataloader whilst two trials are set in the validation dataloader. This was done within each subject to ensure every subject was a part of both training and validation datasets.

4.5.2 Model Architecture

Handling Subject Variability and Initial Projection

The differences between subjects present a great challenge when working with EEG-based decoding. These differences occur due to discrepancies between skull thickness, cortical folding, and exact position of the electrodes during the measurements which means that the same semantic stimulus can elicit spatially distinct EEG patterns across different subjects. To address this, the architecture implemented two different initial projection strategies to map the 44-channel EEG input (C = 44) into the networks hidden dimension (D_{hidden} = 320).

Baseline Approach (Subject Embedding): In the baseline approach, the EEG signals were not actively aligned before processing. The network was instead informed of which subject the data belongs to. This was achieved by initialising a learnable 44-dimensional numerical signature (a subject embedding) that was optimised alongside the network during training. The dimension of this embedding was specifically chosen to match the 44 spatial EEG channels, ensuring a balanced feature representation. This signature was concatenated with the subject's EEG data, and the resulting 88-channel input was then processed by a single, shared 1×1 convolutional layer. This strategy tested whether the deep network could implicitly learn to compensate for anatomical

differences just by knowing the subject's identity.

Using subject-specific spatial filters: Instead of a shared projection, this approach gave each subject their own dedicated projection layer (1x1 convolution). Because skull shapes vary between people, a shared layer might struggle to find common patterns. This method explores whether the network could learn exactly how to mix the 44 electrodes for that specific person. This translates each subject's EEG into a standardised shared latent space. As a result, the deeper layers of the network could focus on decoding semantic content, rather than trying to compensate for the anatomical differences.

Regardless of the chosen handling of subject-identity: an initial 1×1 convolution expanded the representation to a hidden dimension of 320, followed by Batch Normalisation and a GELU activation function.

Temporal Feature Extraction

The main temporal feature extraction was performed by five blocks of dilated convolutional layers as seen in Figure 4.2. Each block incorporates a skip connection that adds the block's input directly to its output. This residual architecture facilitates uninterrupted gradient flow during backpropagation, which is essential for training deep networks without performance degradation. Within these blocks, dilation is used to capture long-range semantics instead of using pooling to avoid losing temporal resolution. The dilation rate increases exponentially across the consecutive blocks ($d \in \{1, 2, 4, 8, 16\}$). This exponential increase provided the network with a broad receptive field, allowing it to integrate extended phonetic and semantic contexts from the EEG signals.

Regularisation

Deep learning models trained on high-dimensional and noisy EEG data are prone to overfitting. To avoid this, channel dropout was used. Several different dropout settings were tested in order to find a balance between overfitting (learning the noise) and underfitting (not learning any patterns of the signal). Furthermore, LayerScale and Batch Normalization were utilised within the

residual blocks to stabilise gradient flow and deep network training. LayerScale introduces a learnable diagonal matrix at the output of each residual block, which is initialised to a near-zero value. By scaling the residual transformations before addition, this mitigates early-stage gradient instability and enables the training of a deeper architecture (Touvron et al., 2021).

An additional experiment was done to test the model's sensitivity to the dimensionality of the semantic targets. PCA was applied to the Whisper embeddings, and the network was trained and evaluated across varying dimensional spaces, specifically testing
$D \in \{1024, 512, 256, 160, 128, 64, 10\}$.

4.5.3 Evaluation Metric

For evaluation, the retrieval task described in the background section was used. A retrieval pool was constructed using the unseen validation dataset. This yielded a candidate pool of 2714 unique, non-overlapping 3-second segments extracted from the 46 trials in the validation dataset.

The retrieval task was performed using cosine similarity to rank the predicted vector against all the candidates. The metrics top-1, top-5, and top-10 accuracy were then defined from this ranking. Given the validation pool size of $N = 2714$ windows, the theoretical baseline for random top-10 retrieval is calculated as $10/N$, yielding approximately 0.37%.

An analysis of the rank distribution was done by producing a rank-histogram to visualise the model's ability to place the correct matching segment towards the top of the list with higher similarity. This ensures that the performance can be understood even if the matching segment does not achieve a top-10 ranking.

4.6 Model 2: A Hybrid CNN-Transformer Encoder

This model was designed with the aim to map neural activity into a shared semantic space with speech representations (Whisper embeddings). To ef-

fectively process the high-dimensional EEG signals, the model used a hybrid architecture that integrates Convolutional Neural Networks (CNN) and Transformers. The choice of a CNN was motivated by its capacity for spatial filtering and ability to extract local correlations in the 44 EEG channels, identifying important signal features. While the CNN captured the local patterns, decoding continuous speech required an understanding of long-range dependencies and context. This motivated the inclusion of a Transformer encoder. Furthermore, the chosen architecture for this model was inspired by the architecture used by d'Ascoli et al. (2025) to decode individual words from EEG. While the architecture in this thesis share foundational similarities with the deep learning pipeline proposed by d'Ascoli et al. (2025), utilising a CNN for feature extraction combined with a Transformer, the two approaches differed in execution and objective, d'Ascoli et.al employed contrastive learning at a sentence level to isolate and decode discrete semantic representations of individual words. In contrast, our CNN-Transformer hybrid mapped continuous, ongoing EEG signals directly onto Whisper's speech representations, processing the data as a continuous stream instead of cutting it into separate words.

The objective for our hybrid model was to investigate the combined CNN and Transformer Encoder, and its accuracy in extracting semantic information. The overall model architecture is presented in Figure 4.3.

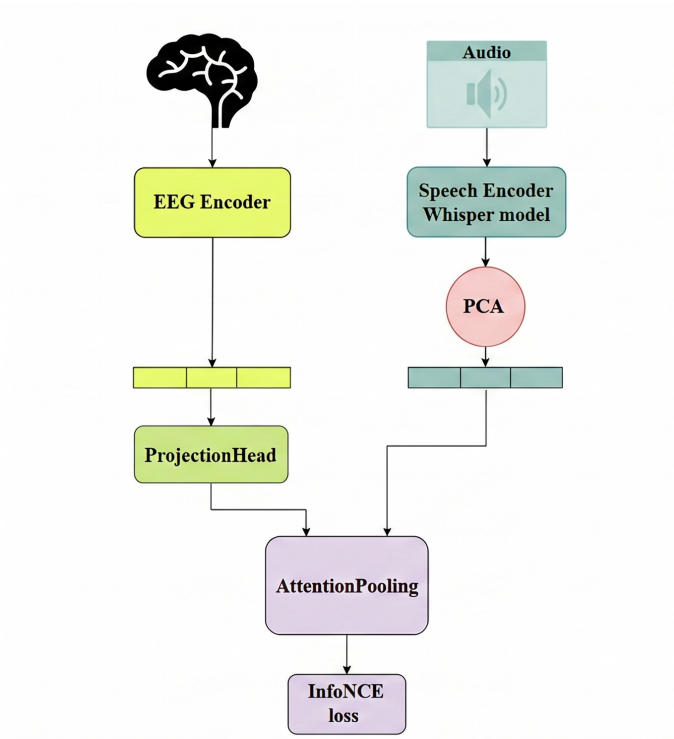


Figure 4.3. Model architecture for model 2.

4.6.1 Data Preprocessing and Data Loading

After the EEG epochs had been created as previously described in the preprocessing step, the EEG had to be matched with the corresponding audio segment. We created an index map providing start and end times for each window, mapping correct audio file to each EEG representation.

To achieve computational feasibility and to optimise model training, an Incremental PCA was implemented as an information bottleneck, reducing the 1024-dimensional Whisper embeddings to approximately 160 components, preserving approximately 80% of the variance. Incremental PCA differs from the standard PCA by processing data in batches rather than all at once, thereby significantly reducing the memory required for computation (Lippi and Ceccarelli, 2019). For comparison, the model was also trained using the

full 1024-dimensional space to evaluate the specific impact of dimensionality reduction on overall performance.

Due to the structure of the dataset and the limited number of total trials, achieving a data split with the exact 80/10/10 distribution was not feasible. Instead, the dataset was partitioned with the strict requirement that all 23 subjects must be represented in both training and validation sets. To achieve this, the data split was performed based on a predefined list of audio trials. However, because the subjects had listened to a randomised selection of audio files, this setup introduced a trade-off.

Maintaining a subject representation across sets and keeping the audio stimuli completely independent was unattainable. As a result, identical audio files appeared across different splits, leading to a degree of data leakage.

These constraints directly impacted all data splits (training. validation, and test). For the validation dataset this resulted in a total of 36 trial-subject pairs built from a pool of 11 unique audio files, meaning multiple subjects shared the same audio stimuli during evaluation. Given that each audio file lasted 178 seconds, segmenting the data into 3-second windows yielded 59 embedding windows per trial (178s / 3s). For the 36 trial-subject pairs, this resulted in a final validation set of exactly 2124 evaluation samples (59×36).

The data were segmented into 3-second windows using an index map, with the PyTorch DataLoader class employed for batch organisation. Due to hardware memory constraints, the physical batch size was limited to 256 samples, however, by implementing gradient accumulation over two steps, an effective batch size of 512 was achieved during training. As previously described in Model 1, the identical segmentation of the two different embeddings, and the parallel indexing, ensured correct alignment with no mismatch between the audio and the EEG.

4.6.2 Model Architecture

The EEG encoder that was designed to extract semantic representations from EEG comprised a CNN encoder as well as a Transformer Encoder. The encoder architecture is illustrated in Figure 4.4.

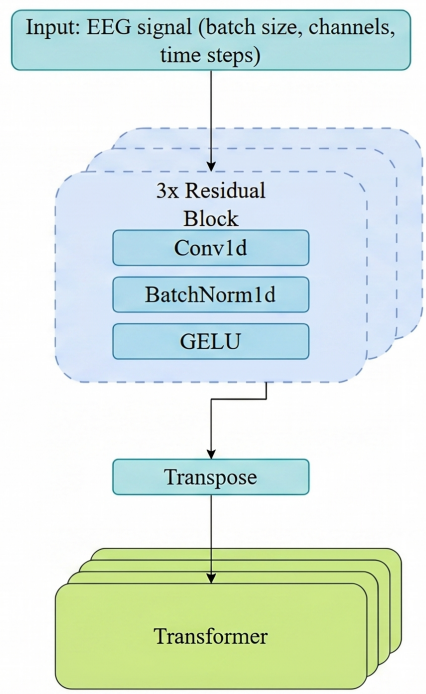


Figure 4.4. Encoder structure of model 2, using a CNN Encoder with 3 residual blocks, and a Transformer Encoder.

CNN Encoder

The initial stage of the EEG Encoder consisted of a 1D Convolutional Neural Network (CNN) designed to extract spatio-temporal features from the EEG signals. The architecture of the CNN block consisted of three residual convolutional blocks, adding shortcuts that enabled improved feature extraction and facilitated gradient flow (Luo et al., 2023; Gori et al., 2023). Each residual block held batch normalisation, a Gaussian Error Linear Unit (GELU)

activation function and dropout of 0.1. Batch normalisation allowed a higher learning rate during training (Ioffe and Szegedy, 2015) meanwhile dropout reduced overfitting of the network (Srivastava et al., 2014). The convolutional blocks expanded the 44 EEG channels to a feature representation of 256 distinct feature maps.

Transformer Encoder

The Transformer used an encoder-only Transformer architecture consisting of 4 layers that analysed the 150 time steps, creating a contextualised output vector. To each layer, 8 parallel attention layers were applied and we used a dropout of 0.2.

Projection and Pooling

To optimise the objective of the contrastive loss function, the EEG embeddings were passed through a non-linear projection head, projecting the embeddings into a shared dimensional space of the same dimension as the reduced Whisper embeddings. The ProjectionHead is a Multilayer Perceptron (MLP) comprising a linear function followed by a non-linear GELU function, Batch Normalisation as well as dropout to improve the generalisation of the model (Xue et al., 2024).

To preserve as much semantic information as possible, attention pooling was chosen over other pooling alternatives. Attention pooling allowed the model to measure how much each time step contributed to the semantic representation, effectively prioritising speech-relevant neural features over background noise (Er et al., 2016). When computing the attention weights, a scaling factor of 2 was introduced to sharpen the attention distribution, forcing the model to be more selective and assign higher weights to the most important temporal segments in the EEG embedding and Whisper embedding, respectively.

Loss Function

As the EEG-embeddings and Whisper embeddings shared the dimension of the Whisper embeddings, an Information Noise-Contrastive Estimation (InfoNCE) loss function was applied for representation learning. InfoNCE is

a contrastive loss function with the goal of maximising the mutual information between the existing context and future data points. Instead of trying to predict the exact signal, the InfoNCE model learned to contrast a matching (positive) sample pair from a pool of false matches (negative samples), thereby extracting underlying latent information shared by the input signals. The number of negative samples therefore affected the performance of the InfoNCE (van den Oord et al., 2019). For training of the model the AdamW Optimizer with a learning rate of 3e-5 was used.

4.6.3 Evaluation Metric

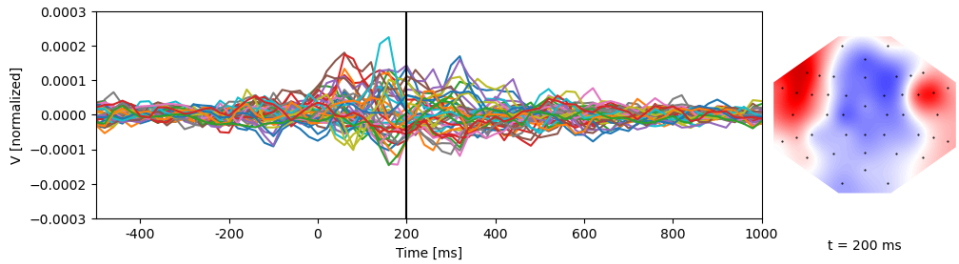
The model was evaluated using unseen validation data to calculate the top-1, top-5, and top-10 accuracy in the retrieval task, as previously described for Model 1. As the data splitting was based on unique audio files for this model, it resulted in a candidate pool of 2124 unique, non-overlapping, 3-second long segments from 11 trials in the validation dataset. Furthermore, a rank-histogram was plotted for this model to examine the model's proficiency in correct segment placement. To evaluate the impact of dimensionality reduction on performance, the validation data was tested under two conditions. Either reduced via PCA to a shared 164-dimensional space, or left unreduced at its original 1024 dimensions. Furthermore, by using the training data for evaluation, the model's learning performance was further investigated.

5

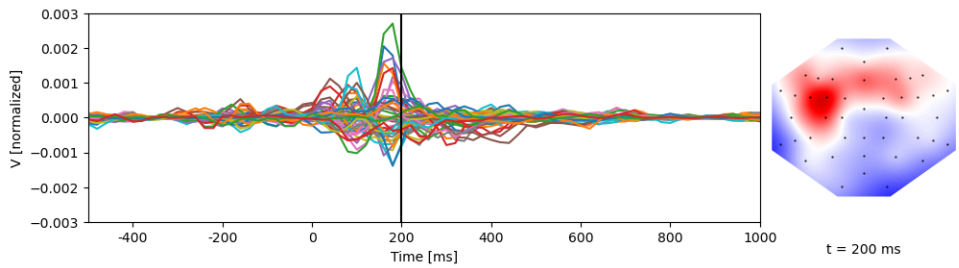
Results and Discussion

5.1 Performance of all TRFs

The neural representations of the predictors were first validated through TRFs. The butterfly plot illustrated in Figure 5.1 displays the TRF weights for the 10 Whisper components in subfigure 5.1a alongside the acoustic envelope (gammatone) in subfigure 5.1b. A perceptible difference in morphology and amplitude is observed. Specifically, the Whisper features yield a lower amplitude and less distinct peaks compared to the high-amplitude response of the gammatone predictor. In the plot displaying the gammatone predictor, expected peaks are visible at ~100 and ~200 ms as previously described in the background section. The TRF of Whisper's contribution lacks these distinct peaks. However, a clear neural response is observed following time 0 for the Whisper features, confirming that these embeddings capture some variance in the EEG signal that is temporally aligned with the stimulus.



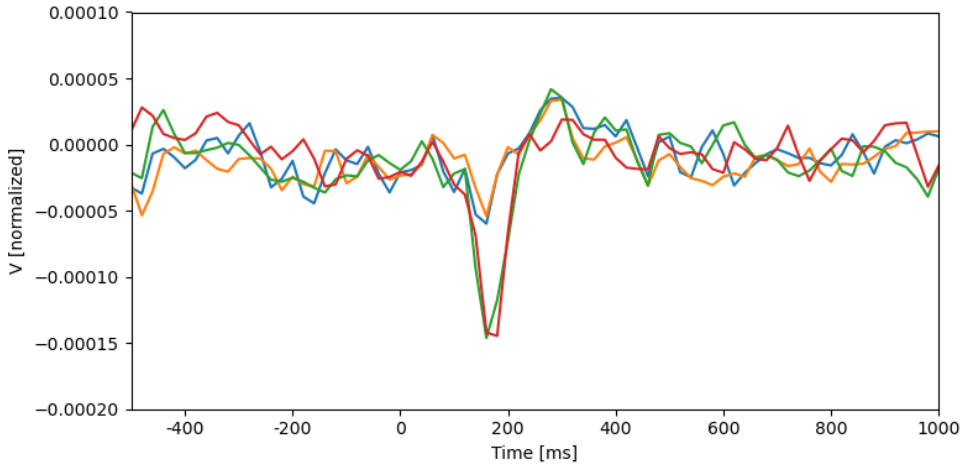
(a) Left: Butterfly plot of the mean of all 10 Whisper features averaged over all 23 subjects. **Right:** The topomap shows the activation of sensors on the scalp at time 200 ms after stimulus onset.



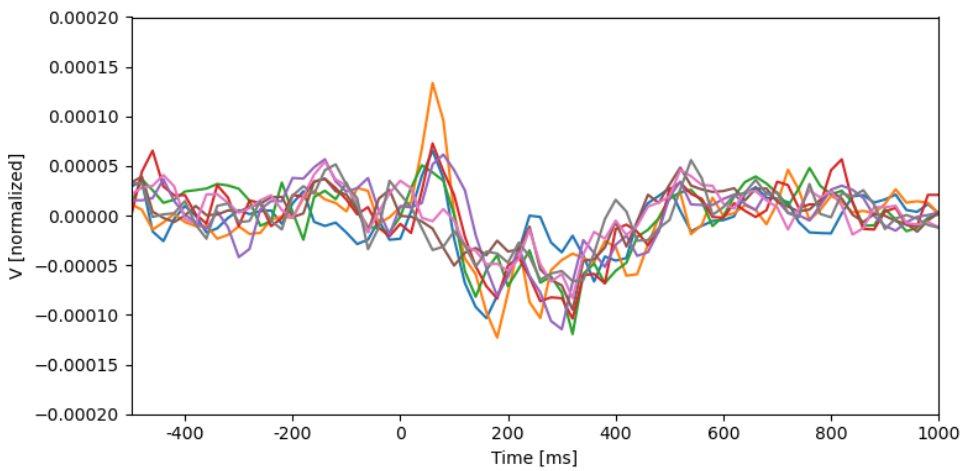
(b) Left: Butterfly plot of the gammatone feature averaged over all 23 subjects. **Right:** The topomap shows the activation of sensors on the scalp at time 200 ms after stimulus onset.

Figure 5.1. TRFs showing how the different predictor variables reconstruct the neural response. The TRFs are averaged over all 23 subjects and each line represent one of the 44 channels marked as black dots on the topomap to the right. Subplot a) present the mean of the resulting TRF of the 10 Whisper features, and subplot b) shows the gammatone predictors contribution to the TRF. The topomaps to the right show the neural activity in the 44 channels at the specified time point (200 ms) marked with a vertical line.

A more distinct visualisation of the peaks present in the TRF for the Whisper features is presented in Figure 5.2. Subfigure 5.2a shows the reconstructed neural response for the frontal channels (Fp1, Fp2, AFz, and Fz), while subfigure 5.2b presents the response for the central channels (FC1, FC2, C1, Cz, C2, CP1, CPz, and CP2). Figure 5.3 illustrates a typical 10-20 EEG sensor montage that shows the placement of the frontal and central sensors mentioned above.



(a) TRF plot of Whisper features for the frontal sensors.



(b) TRF plot of Whisper features for the central sensors.

Figure 5.2. TRFs of the Whisper features for selected sensors. 5.2a displays the TRF in the frontal sensors, and 5.2b displays the TRF for the central sensors.

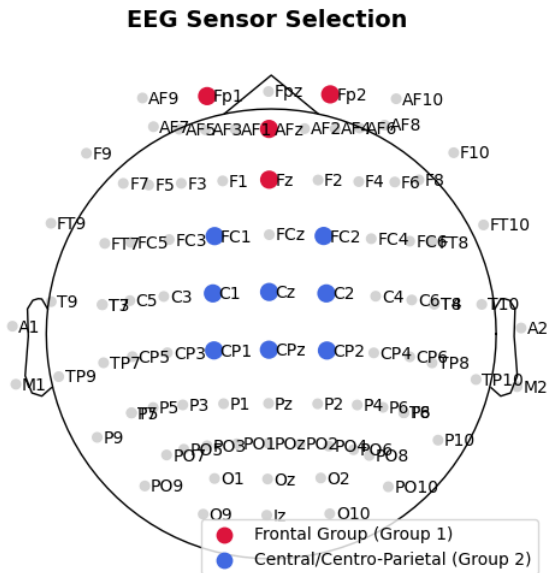


Figure 5.3. Standard 10-20 sensor montage with frontal and central sensors used in Figure 5.2 that are coloured red and blue, respectively.

For the frontal sensors in subfigure 5.2a, a prominent negative deflection is visible, followed by a smaller positive peak. These features are interpreted as potentially corresponding to the N1-P2 complex described in Section 2.3.1, which are linked to stimulus onset and auditory attention. Given that both the N1 and P2 components are typically most prominent in the frontocentral brain regions, this alignment suggest that the TRF captures a characteristic neural response to audio.

The TRF for the central sensors in subfigure 5.2b displays a slightly different pattern, characterised by an early positive peak followed by a negative deflection, suggesting a P1-N1 complex. Taken together, these TRF morphology variations indicate the presence of typical auditory processing and attentional modulations.

Correlation and Model Performance

The efficacy of these predictors is further supported by the Pearson correlation analysis presented in Figure 5.4. These results demonstrate a clear

hierarchy in model performance (*Gammatone + true Whisper* > *Gammatone* > *Gammatone + false Whisper*). The fact that the inclusion of Whisper features increases the Pearson correlation across almost all channels (with the exception of F3) suggests that Whisper captures linguistic information that is complementary to the acoustic envelope. Furthermore, the degraded performance of the null model (*Gammatone + false Whisper*), where linguistic features were misaligned, confirms that the TRF framework is sensitive to the precise temporal alignment of semantic content.

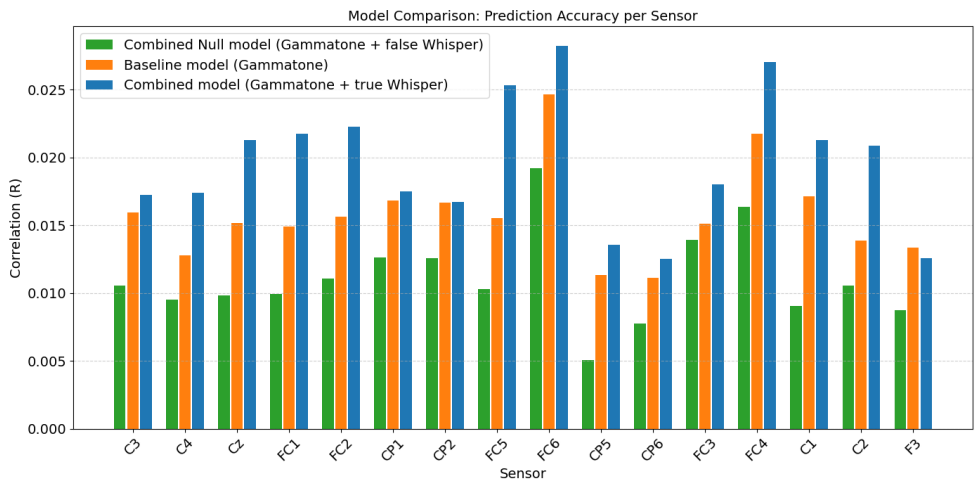


Figure 5.4. Prediction accuracy per sensor measured in correlation (r). The plot compares the of three different feature sets: a combined model using Gammatone and false Whisper, a baseline Gammatone model, and a combined model using Gammatone and true Whisper features.

Connection to the Deep Learning Models

The TRF results put the primary objective of this thesis in an interesting perspective. While the TRF analysis shows that Whisper features are "weaker" neural predictors than the envelope when using linear modelling, the correlation increase in Figure 5.4 proves that they contribute unique information that the envelope lacks.

TRFs serve as an initial validation that relevant information is present in the

EEG. However, since TRFs are linear, they do not directly predict nonlinear model performance.

PCA and Computational Constraints
The results using the 10-component PCA model indicate that even a highly compressed representation of Whisper embeddings preserves the variance connected to auditory processing. During the experimental phase, it was observed that increasing the number of PCA components led to an increase in training time that exceeded the available computational capacity. Thus, the 10-component model represents a necessary trade-off between semantic depth and computational feasibility.

5.2 Evaluation and Comparison of Model Architectures

This section presents the results of the two distinct EEG encoder architectures, followed by a comparative analysis of their performance.

Before evaluating the metrics, an inherent limitation regarding the data split must be addressed. Because the data was divided based on trials, there is a stimulus overlap where identical audio segments occur across both the training and validation sets. Consequently, the reported results should be interpreted as the model's performance on previously encountered semantic representations, rather than strict generalisation to new data.

5.2.1 Model 1: Dilated CNN Encoder

Impact of Dimensionality and Regularisation
Figure 5.5 plots the top-10 retrieval accuracy on the y-axis against the target dimensionality (number of PCA components) on the x-axis, with separate curves representing different dropout rates. Table 5.1 provides the corresponding exact numerical percentages for these configurations.

As illustrated in Figure 5.5 and Table 5.1, the relationship between target dimensionality and retrieval accuracy follows an inverted U-shape across all regularisation levels. The network achieves its peak performance at a

reduced target space of 64 dimensions, where a dropout rate of 40 % yields the highest top-10 accuracy of 4.50 %. This means that the true matching sequence was ranked among the top 10 candidates for 4.50% of the evaluated windows. Compared to the theoretical random baseline of 0.37 % for this specific candidate pool (2714 windows), this optimal configuration performs approximately 12 times better than random chance. This indicates that the model can extract features from the EEG data that correlate with the target semantic representations.

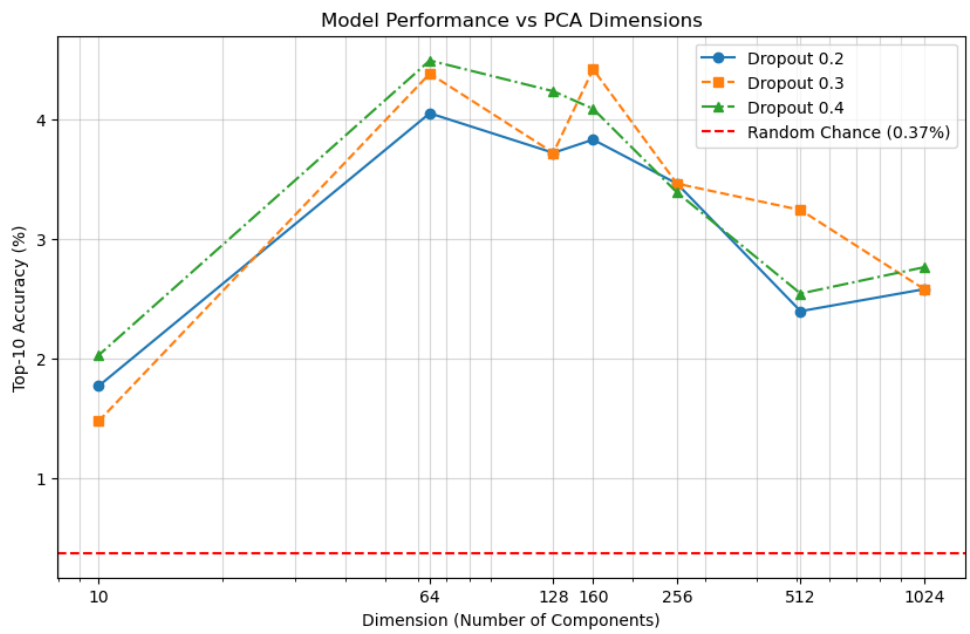


Figure 5.5. Top-10 accuracy across different PCA dimensions evaluated for three different dropout rates (0.2, 0.3, and 0.4). The red dashed line represents the random chance baseline at 0.37 %.

However, performance decreases at both extremes. Extreme compression (PCA 10) likely discards essential semantic features, resulting in the lowest accuracy scores. On the other hand, attempting to map to the full, uncompressed Whisper embeddings (1024 dimensions) introduces excess variance that the network seems to struggle to map, causing a drop in performance

Table 5.1. Top-10 retrieval accuracy with different dropout levels.

Dimension	20 (%) dropout	30 (%) dropout	40 (%) dropout
1024	2.58	2.58	2.76
512	2.39	3.24	2.54
256	3.46	3.46	3.39
160	3.83	4.42	4.09
128	3.72	3.72	4.24
64	4.05	4.38	4.50
10	1.77	1.47	2.03
Random Chance	0.37	0.37	0.37

across all dropout rates.

This performance drop at higher dimensions lead to an interesting reflection when compared to our preliminary validations. The initial TRF analysis indicated that the EEG signals contained trackable responses to the stimulus, but this linear mapping was only successful when the target embeddings were heavily compressed (PCA 10). Expanding the target space from 10 to the full 1024 dimensions introduces a massive increase in complexity and variance.

For this specific convolutional architecture, compressing the target down to 64 dimensions appears necessary to filter out what the network perceives as noise, allowing it to focus on core semantic features. However, this raises an important question: does the model fail at 1024 dimensions because the EEG signal inherently lacks the information required to map such a complex space, or is it simply a limitation of this specific architecture's capacity to integrate high-dimensional targets?

Evaluating Subject-Specific Spatial Projections

To isolate the architectural impact of spatial filtering from general hyperparameter tuning, the optimal configuration identified in the previous section (Dropout 0.4, PCA 64) was locked in as the baseline. Using this fixed capacity, the performance of the shared 1×1 projection layer was compared against the architecture utilising distinct, subject-specific spatial filters.

As seen in the Table 5.2 it is shown that the architecture utilising subject-specific spatial filters yielded a lower top-10 accuracy than the shared projection baseline. Given the significant anatomical differences between individuals, the original assumption was that personalised initial projections would be necessary for the network to accurately map each subject's unique EEG patterns.

Table 5.2. Comparison of top-10 Retrieval Accuracy across spatial projection strategies. Both models utilize the established optimal baseline configuration (Dropout 0.4, PCA 64).

Spatial Projection Strategy	**Top-10 Accuracy (%)**
Shared Projection (Baseline)	5.34
Subject-Specific Filters	4.50

This result is likely due to data scarcity at the individual subject level. Each subject's dataset is relatively small, consisting of seven trials (with 59 three-second segments per trial). This limited volume of data restricts the network's ability to effectively optimise a unique set of spatial weights for each person. On the other hand, the shared projection layer benefits from aggregating data across all subjects. This provides a substantially larger training pool, allowing the network to learn more robust and generalised spatial mappings that implicitly compensate for anatomical variations. This suggests that, under limited per-subject data, shared representations outperform subject-specific parameterisation.

Retrieval Task Performance

As previously established, the evaluation setup contains inherent stimulus-data leakage. Furthermore, because multiple subjects listened to the same audio files, identical audio segments appear several times within the validation dataset and the retrieval pool. This creates a challenging retrieval dynamic. The network might correctly identify the semantic representation but assign a high similarity score to an identical "twin" audio clip. This can artificially push the strictly correct target further down in the ranking. Because of

this, a strict top-10 metric might underrepresent the model's actual ability of mapping to the target space.

To provide a clearer view of the model's performance, rank distribution histograms for both the trained model and the untrained baseline are presented in Figures 5.6 and 5.7. These distributions illustrate exactly where the correct target was placed within the candidate lineup. The x-axis represents the target's rank (from 1 to the total number of candidates), divided into 100 bins. Consequently, each bar represents a range of approximately 27 rank positions. The y-axis shows how many evaluated EEG windows resulted in that specific rank.

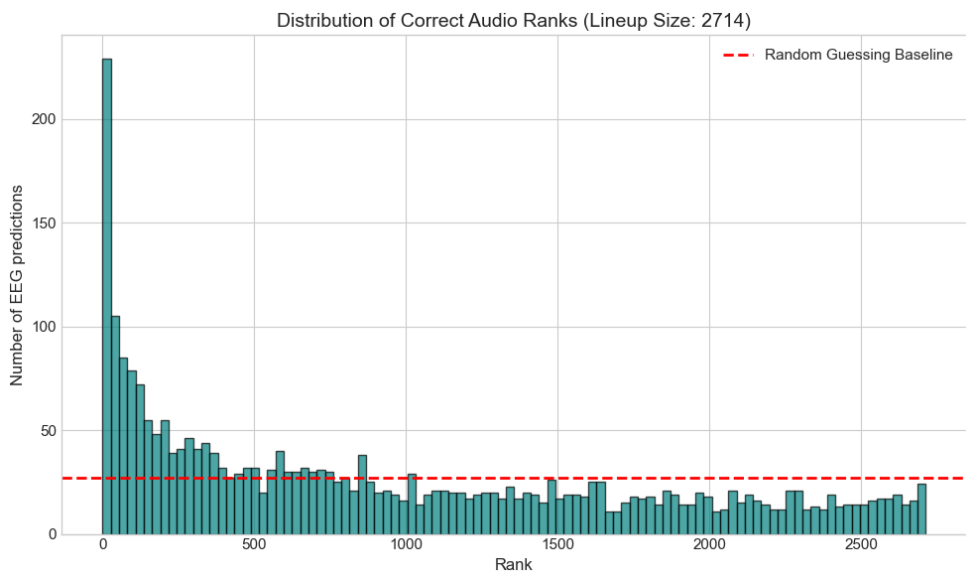


Figure 5.6. Rank distribution from the retrieval task with 64 dimensions. A shift in the distribution towards rank 1 indicates that the correct segments were successfully assigned high similarity scores.

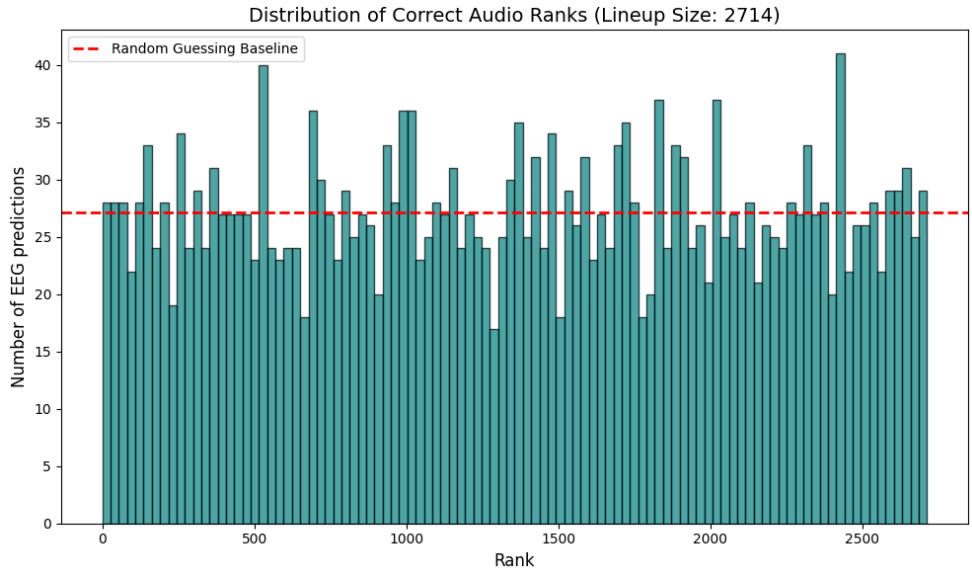


Figure 5.7. Rank distribution of an untrained baseline model. The x-axis represents the rank assigned to the correct target audio among the full candidate pool (divided into 100 bins), and the y-axis shows the frequency of occurrences. Random guessing results in a uniform, flat distribution.

The trained model shows a clear shift toward the lower ranks. As seen in the histogram in Figure 5.6, for more than 200 evaluated EEG windows, the correct match is placed within the very first bin, ranking it among the most similar candidates. This creates a clear downward slope, demonstrating that the model performs significantly better than random guessing. In contrast, the rank distribution of the untrained baseline is uniform, as demonstrated in Figure 5.7, meaning the network randomly places the correct match anywhere within the lineup.

Even when the correct segment is not retrieved within the top-10, it is consistently placed among the top-ranking candidates, indicating partial semantic alignment. By successfully pushing the correct semantic targets to the top of the similarity ranking, the model confirms its ability to capture the underlying semantic representations of the segments.

To summarise, the evaluation of Model 1 suggests a connection between the model's architecture, the size of the target space, and the amount of available data. The inherent complexity of EEG signals makes this type of mapping challenging, and the local filters of the convolutional network seem to struggle with disentangling the full 1024-dimensional Whisper space. This is likely why reducing the target to 64 dimensions appears necessary to filter out complex noise. Furthermore, the limited amount of training data per subject likely makes personalised spatial filters ineffective, meaning the model must rely on a shared projection layer to combine data across all subjects. Although the absolute performance metrics remain low, the results show a clear improvement over random guessing. This suggests that Model 1 succeeds to a reasonable degree in identifying semantic patterns, but the architecture likely requires a smaller hypothesis space and shared data to learn effectively.

5.2.2 Model 2

The Effect of Dimensionality on Model Performance and Retrieval Task Accuracy

Table 5.3 presents the retrieval task scores of the validation data for models trained both with and without PCA implementations.

Table 5.3. Comparison of top-K retrieval accuracy between the model trained with dimensionality reduction (160 principal components, retaining approximately 80% of the audio variance) and the model trained without dimensionality reduction.

	Top-1(%)	**Top-5(%)**	**Top-10(%)**
Random chance	0.05	0.24	0.47
PCA (164 dim)	0.42	3.95	6.54
No PCA (1024 dim)	1.79	8.24	12.43

The results demonstrate a substantial performance increase over random chance for both approaches, with the unreduced model consistently outper-

forming the PCA-reduced alternative. Specifically, the model trained with dimensionality reduction achieved top-1, top-5, and top-10 accuracies that were approximately 8x, 16x, and 14x better than random chance, respectively. This reduction in dimensionality significantly decreased the computational load while retaining the most essential features of the acoustic representations. In comparison, omitting dimensionality reduction yielded a much stronger performance with top-1, top-5, and top-10 accuracies surpassing random chance by factors of roughly 36x, 34x, and 26x.

The removal of the PCA bottleneck and the transition to the full 1024-dimensional Whisper embeddings yielded a considerable increase in retrieval performance. This performance improvement is presumably due to a combination of the loss function benefiting from a higher-dimensional space and the attention mechanisms present in the Transformer architecture and the attention pooling.

The 1024-dimensional alignment space allowed the InfoNCE contrastive loss to provide highly nuanced feedback to the EEG Transformer's attention mechanism as the weights were modified to match the Whisper embeddings. Because Whisper is also built on a Transformer architecture, its embedding space naturally stores highly detailed speech information. By preserving all 1024 dimensions, the EEG Transformer is believed to have had the capacity to optimise its attention weights to better represent the information in the Whisper embeddings.

To gain a deeper insight into the cross-modal alignment between Whisper embeddings and the EEG features, the model performances were further evaluated using rank histograms. With a total of 2124 evaluation samples distributed across 100 bins, each bin represents approximately 21 rank positions, with the first bin capturing the highest-ranking retrievals. Figures 5.8 and 5.9 display the rank distributions of the 164-dimensional PCA model and the uncompressed 1024-dimensional model, respectively. Notably, the full-dimensional model demonstrates a significant increase in the first bin, indicating a substantially higher rate of top-21 rankings.

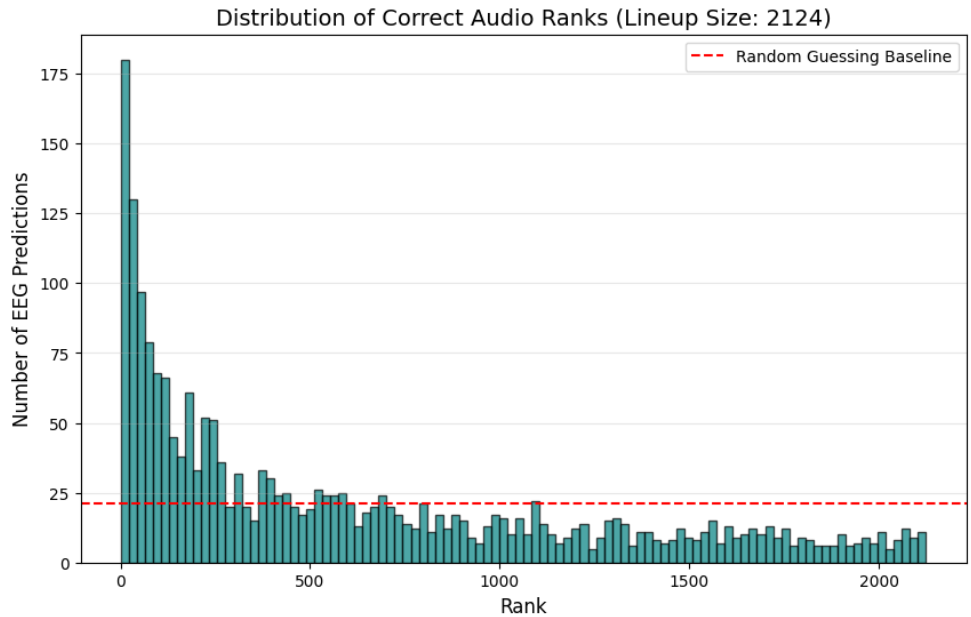


Figure 5.8. Rank histogram of the results from the retrieval task with 164 dimensions.

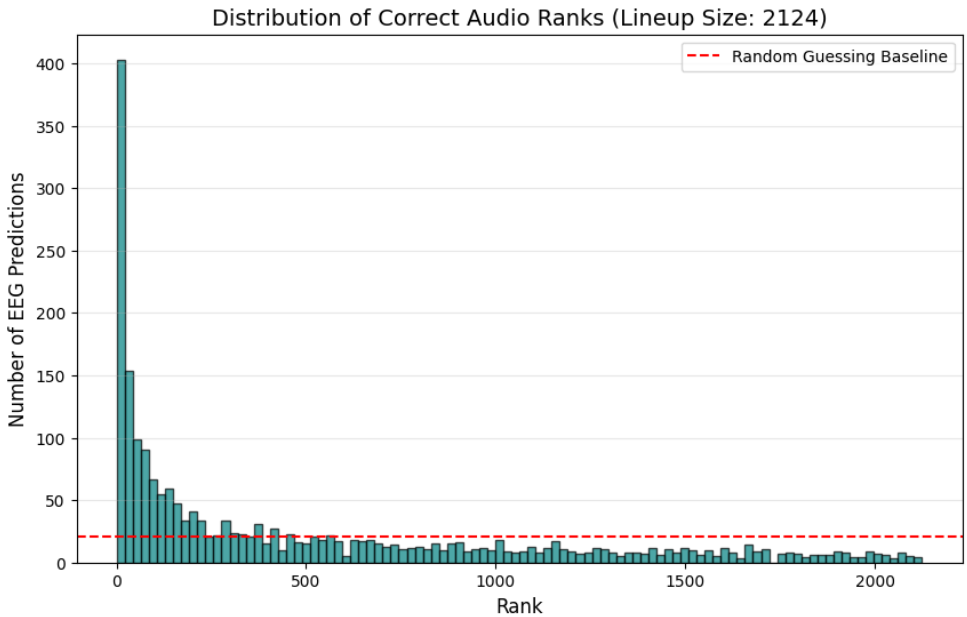


Figure 5.9. Rank histogram of the results from the retrieval task with all 1024 dimensions.

The evaluation of the validation dataset reflects a complex interaction between two counteracting biases. On one hand, the stimulus leakage from the training data likely led to an overestimation of the model's absolute performance, as the network had already encountered the acoustic-semantic representations of the validation data. On the other hand, the retrieval metrics and rank histograms may have suffered from an underestimation due to target collision within the evaluation batch. Because the validation dataset contained duplicate audio files across multiple subjects, identical audio segments produced identical Whisper embeddings. During the contrastive ranking process, a subject's EEG embedding yielded identical similarity scores against these duplicate Whisper targets. If the ranking algorithm placed these identical scores ahead of the subject-specific pair, the ranking position would be depressed.

Consequently, while the model successfully identified the correct semantic content, the presence of duplicate targets created a ranking ambiguity that understates the model's true retrieval performance. The results in Table 5.3

and Figure 5.8 & 5.9 should therefore be interpreted with both of these factors in mind.

Retrieval Performance on the Training Dataset

To examine how well the model learned from the training data and detect eventual overfitting, the training data was used for evaluation. Table 5.4 presents the top-K accuracies from the retrieval task and Figure 5.10 illustrates the rank histogram across all 11505 training samples. The rank histogram show a similar shape to that for the validation data in Figure 5.8, which further strengthens that the model has learned to recognise generalisable semantic features instead of just memorising the training data.

Table 5.4. Top-K accuracies from retrieval task using training data with 164 dimensions.

	Top-1 (%)	**Top-5 (%)**	**Top-10 (%)**
Random chance	0.01	0.043	0.09
Training data (164 dim)	0.34	1.52	2.71

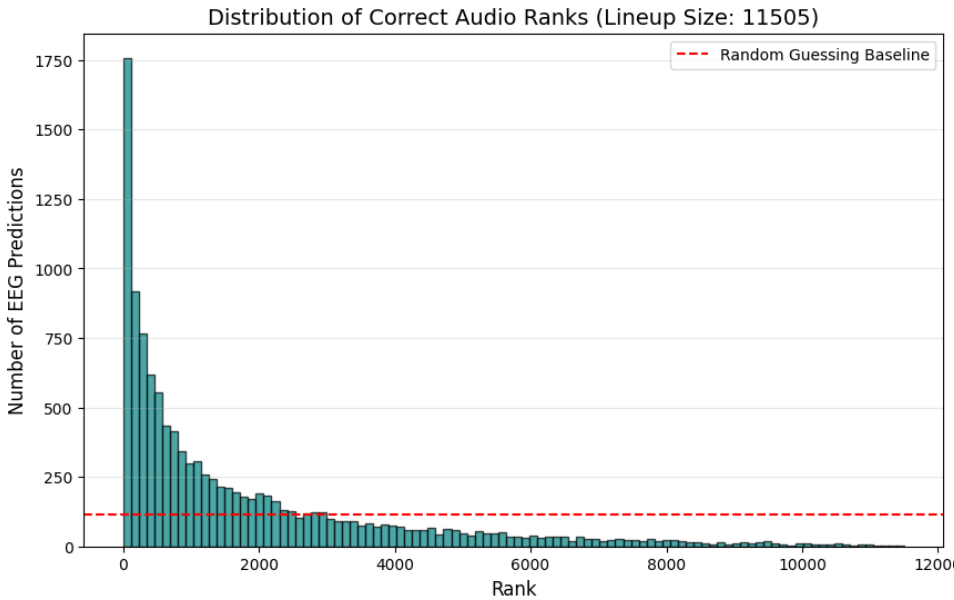


Figure 5.10. Rank histogram of the retrieval task when the training data (164 dimensions) was used for evaluation.

To enable a fair evaluation between the training and validation datasets, the retrieval performances were normalised against their respective random chance levels. Since the datasets differ significantly in size, their baseline random chances for the top-10 accuracy vary accordingly (0.47% for validation vs. 0.09% for training). Rather than comparing absolute percentages, the performance was expressed as a fold-increase over the baseline chance level. The results demonstrate that while the validation model achieves a top-10 accuracy of 6.54% (representing a 13.9 fold increase over chance), the training retrieval reaches 2.71%, which corresponds to a 30.1 fold increase over its respective chance baseline.

These results indicate an expected generalization gap. However, since the performance of the unseen validation data is ~14x better than the random chance, it implies that the model has extracted robust, generalisable features rather than being highly overfitted.

The top-K results presented in Table 5.3 and 5.4 are, at first sight, quite low. It is valid to ask whether a top-10 accuracy of 12.43% is enough to imply that the model has can extract semantic information from EEG signals. EEG signals are highly noisy and complex data, filled with artifacts from muscle movement, blinks, and external interference, making continuous speech decoding a difficult task. However, when comparing the accuracy to the random chance, we get a better understanding of the model performance, proving that the model is not guessing randomly, but has indeed learned to identify meaningful semantic patterns within the noisy EEG data.

5.2.3 Model Comparison

The key difference between the models lies in their ability to operate in high-dimensional target spaces. The results showed a clear contrast in this regard: the hybrid Transformer architecture performed better when using the full 1024-dimensional Whisper space. In contrast, the convolutional baseline achieved its best results when the targets were reduced to 64 dimensions. Table 5.5 presents a comparison of top-10 accuracy versus random chance for each architecture's optimal configuration. The best-performing Model 1 framework achieved a 12-fold increase over random chance in the top-10 retrieval test, whereas the optimal configuration of Model 2 demonstrated a substantially higher performance, yielding a 26-fold increase.

Table 5.5. Fold-increase comparison for top-10 accuracy versus random chance between the optimal configurations for each model architecture.

Model Configuration	Dimensions	Top-10 (vs. Random)
Model 1 (CNN)	64	~12x better
Model 2 (CNN+Transformer)	1024	~26x better

As explained in Section 2.4.4, this difference can be understood through the models' inductive biases. The weaker inductive bias of a Transformer gives it an advantage when mapping to a complex and high-dimensional space. On

the other hand, the strong inductive bias of a CNN means it needs a smaller hypothesis space to learn effectively from the available data.

Another possible explanation for this performance gap relates to how the target data was created. The target space consists of embeddings from Whisper, which is also a Transformer-based model. Because self-attention mechanisms shape this target space, an EEG model that uses the same architecture might have a natural advantage. This shared structure could make it easier for Model 2 to map neural signals into the full 1024-dimensional space, compared to model 1, which is built mainly with a CNN that uses a different mathematical approach.

Furthermore, while the CNN has tools to understand broader context, this does not solve the issue with high dimensions. Model 1 uses dilated convolutions to expand its view, allowing it to integrate information over longer time windows. However, being able to look at a longer timeframe is not the same as having the capacity to handle 1024 different features at once. A reasonable hypothesis is that while dilation helps the CNN see the global context, its local filters struggle to manage the complex gradients of the full target space. This limitation is likely what makes the reduction to 64 dimensions necessary.

5.3 Limitations

Data split

To evaluate the architectures, two different data splitting strategies were employed. For Model 1, the dataset was split sequentially by subject, assigning the first seven trials to the training set and the final two trials to the validation set. In an attempt to control the distribution of audio segments and reduce overlap, the data for Model 2 was divided based on a predefined list of specific stimuli (a stimulus-based split).

However, due to the limited total number of unique audio files available in the dataset, achieving a strict, non-overlapping split was not feasible for either model. While the sequential split for Model 1 naturally caused overlap

because subjects listened to different trial combinations, Model 2 faced a data volume constraint. Even when intentionally separating the data by stimulus, the restricted pool of unique audio meant that identical files still had to be reused across both the training and validation sets.

Because of these constraints, identical target embeddings appear multiple times within the validation set for both models, creating a challenging dynamic for retrieval accuracy. The ideal data split would have strictly isolated the audio stimuli between the sets to guarantee zero overlap, while simultaneously ensuring every subject was represented in both. Due to the high number of trials and time constraints during the project, this perfect separation was not feasible.

Because the EEG signals were unique to each subject but their corresponding target embeddings were sometimes identical, this overlap may lead to a misestimation of the models' true generative capabilities to entirely unseen data. An analysis of the validation set revealed that a single audio file could appear up to four times. However, because a top-10 retrieval metric is used to evaluate the models, the negative impact of these recurring "twin" targets is significantly mitigated. The histogram also shows how the model performed which is an important plot to keep when having this flaw since the correct segment might have been pushed slightly outside the top-10 but should either way stayed close to the most similar segments ranking.

Batch Size

Due to local hardware limitations (8GB RAM), early training iterations were restricted to a small batch size. However, this caused significant instability during training due to the network's Batch Normalisation layers. Because Batch Normalisation relies on calculating the mean and variance of each individual batch, a very small batch size leads to highly volatile and unreliable statistics that do not represent the overall dataset. This instability prevented the network from converging. To resolve this and ensure stable normalisation, the training process was migrated to a more capable computing resource, which allowed for a sufficiently large and optimal batch size.

Computational Constraints

Due to lack of GPU power and the time constraint the models were trained once. For a more optimal result the model would have been trained using K-fold cross validation and trained several times to obtain an average between the runs to be used as the result. Since this was not possible in this project the result and discussions are based on the bigger trends seen across the results.

Design choices across the models was also from the start adapted to work on a less powerful computer. They were later moved to a more powerful GPU.

Data Scarcity

Of the three available experimental conditions, only one is utilised for training the models in this thesis. As mentioned in the dataset chapter, this specific subset has 23 subjects, with each subject listening to 9 audio trials lasting 178 seconds each. In total, this yields approximately 10 hours of EEG recordings. Applying a standard 70/30 train-validation split results in roughly seven hours of usable training data. Because these seven hours are distributed across 23 different individuals, the actual volume of training data available per subject is highly limited, creating a notable data scarcity constraint at the individual level.

Loss function

Both the baseline architecture and the comparative Transformer model utilise forms of Contrastive Loss (such as InfoNCE) for sequence-level retrieval. While mathematically effective for representation learning, these loss functions are inherently "semantically blind." They rely on index matching when calculating the error.

It treats all the non-matching indexed segments as equally wrong. The network might have predicted a segment semantically similar to the correct segment although it is penalised just as much as if it predicted a segment that has no semantic similarities.

Reliability of Ground Truth

A fundamental challenge in evaluating these models is the absence of an

absolute "ground truth" for the brain's semantic processing. The temporal alignment is definitive, meaning that it is known which EEG segment corresponds to which Whisper embedding. Although this index matching does not guarantee that the embedding is a pure representation of the neural activity.

The TRF analysis was used to get some clarification on this concern. The TRF results show that the Whisper embedding contains information correlated with the EEG signals. However, this correlation was only robust when the Whisper embeddings were used in combination with a Gammatone representation (the acoustic envelope) as predictors. The semantic Whisper embeddings alone were not enough to produce a strong TRF response.

This creates a significant challenge for deep learning architectures in this thesis, as they are trained to map exclusively from the EEG signals to the Whisper space. Because the baseline verification required both acoustic and semantic predictors, it is impossible to perfectly isolate which dimensions of the Whisper embeddings represent the true "semantics" the brain is tracking, and which parts are simply untrackable noise.

5.4 Future work

Computational Power and Computing Time

For future work real-world applications like hearing aids, the computational power and processing time of the model must be addressed. Currently, the architecture requires more than the 8 GB of RAM typically found in standard laptops. Combined with long computational times, this is a major challenge for wearable devices. Future work should therefore focus on optimising the model size and speed, making it lightweight enough to run efficiently on low-power hardware.

Whisper Layers and Model Size

This experiment is based on the assumption that the final layer of the Whisper model contains the most detailed contextualised representations. However, because the precise nature of the linguistic and acoustic information captured

within these latent spaces remains an open question, further investigation into the Whisper architecture would be highly beneficial. Future work could propose a more profound analysis of Whisper's internal layers to determine their individual contributions to speech representations. Additionally, it would be valuable to explore how varying the Whisper model size impacts overall performance. While the current architecture utilises the relatively complex Whisper-medium variant (1024 dimensions), evaluating both less complex and more scaled-up versions within the same pipeline would provide crucial insight into model scaling and representation efficiency.

Optimise Loss function for Semantic learning

As identified in the limitations, the current contrastive loss functions optimise the latent space using a binary matching system, failing to account for semantic similarities between different audio segments. Given that the primary goal of this research is semantic decoding, addressing this constraint is a good direction for future work.

Future architectures could benefit from implementing a semantically aware loss function. Instead of penalising all incorrect candidates equally, the objective function could incorporate "soft labels" or distance-weighted penalties based on the actual cosine similarity between the text embeddings of the candidates. By rewarding the network for predictions that are semantically adjacent to the true target, even if the exact index is incorrect, the model could learn to map the continuous semantic space more effectively and capture linguistic nuances from the EEG data.

6

Conclusion

This thesis aims to decode semantic-like information from EEG signals by mapping neural activity to a high-dimensional semantic space generated by the Whisper model. As an initial baseline, Temporal Response Function (TRF) models confirmed that the EEG signals contained trackable responses to the stimuli by using the acoustic envelope and the compressed Whisper embeddings as predictors. However, a central challenge in this work is the absence of an absolute ground truth for how the brain represents semantics. Since there is no way to directly observe the brains semantic processing, this work used Whisper's artificial embeddings as a substitute. This creates an uncertainty regarding whether the models are learning to track true human comprehension or just approximating a specific machine learning algorithm.

The results demonstrate a clear difference in capability between the two evaluated models. Model 1 using dilated convolutions struggled to manage the full complexity of Whisper's 1024-dimensional space, requiring a compressed target space of 64 dimensions to filter out noise and converge. In contrast, the transformer based Model 2 successfully mapped neural signals directly to the un-compressed target space. This suggests that Model 2's attention mechanisms share a structural alignment with the Whisper embeddings. Therefore, transformer-based architectures appear to be a more promising direction for

future development when mapping neural data to complex semantic spaces.

The findings must also be interpreted with certain limitations in mind regarding the method. First, the limited amount of training data available per subject made the personalised spatial filters ineffective for Model 1. Second, the evaluation was complicated by duplicate audio segments within the validation set. Because identical targets appeared multiple times, the strict ranking process could not differentiate between them, which likely led to an underestimation of the models' true performance. Finally, the contrastive loss functions applied here lack semantic awareness, meaning they penalise every incorrect prediction equally without considering its linguistic proximity to the true target.

Despite these limitations and the lack of a perfect ground truth, the results demonstrate clear potential. Both architectures performed significantly better than random chance in the retrieval tasks. It shows that it is possible to extract meaningful semantic patterns from EEG recordings.

Bibliography

Anderson, Andrew J., Christopher Davis, and Edmund C. Lalor (2023) "Context and Attention Shape Electrophysiological Correlates of Speech-to-Language Transformation," 10.1101/2023.09.24.559177.

Auster, Quentin, Kateryna Shapovalenko, Chuang Ma, and Demaio Sun (2025) "A Penny for Your Thoughts: Decoding Speech from Inexpensive Brain Signals," 10.48550/arXiv.2511.04691.

Berman, Jules J. (2016) "Chapter 4 - Understanding Your Data," in *Data Simplification*, 135–187, Boston: Morgan Kaufmann, https://doi.org/10.1016/B978-0-12-803781-2.00004-7.

Brodbeck, Christian, Proloy Das, Marlies Gillis, Joshua P Kulasingham, Shohini Bhattasali, Phoebe Gaston, Philip Resnik, and Jonathan Z Simon (2023) "Eelbrain, a Python Toolkit for Time-Continuous Analysis with Temporal Response Functions," *eLife*, 12, e85012, 10.7554/eLife.85012.

Chaddad, Ahmad, Yihang Wu, Reem Kateb, and Ahmed Bouridane (2023) "Electroencephalography Signal Processing: A Comprehensive Review and Analysis of Methods and Techniques," 23 (14), 6434, 10.3390/s23146434.

d'Ascoli, Stéphane, Corentin Bel, Jérémy Rapin, Hubert Banville, Yohann Benchetrit, Christophe Pallier, and Jean-Rémi King (2025) "Towards decoding individual words from non-invasive brain recordings," *Nature Communications*, 16 (1), 10521, 10.1038/s41467-025-65499-0.

Dauphin, Yann N., Angela Fan, Michael Auli, and David Grangier (2017) "Language Modeling with Gated Convolutional Networks," September, 10.48550/arXiv.1612.08083.

Défossez, Alexandre, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King (2023) "Decoding Speech Perception from Non-Invasive Brain Recordings," *Nature Machine Intelligence*, 5 (10), 1097–1107, 10.1038/s42256-023-00714-5.

Deininger, Luca, Bernhard Stimpel, Anil Yuce, Samaneh Abbasi-Sureshjani, Simon Schönenberger, Paolo Ocampo, Konstanty Korski, and Fabien Gaire (2022) "A Comparative Study between Vision Transformers and CNNs in Digital Pathology," https://arxiv.org/abs/2206.00389v1, June.

Drosos, Konstantinos, Alexandra Papanicolaou, Louiza Voniati, Klea Panayidou, and Chryssoula Thodi (2024) "Auditory Processing and Speech-Sound Disorders," 14 (3), 291, 10.3390/brainsci14030291.

Elbayad, Maha (2020) *Rethinking the design of sequence-to-sequence models for efficient machine translation* Ph.D. dissertation, Université Grenoble Alpes, `https://tel.archives-ouvertes.fr/tel-02941570`.

Er, Meng Joo, Yong Zhang, Ning Wang, and Mahardhika Pratama (2016) "Attention Pooling-Based Convolutional Neural Network for Sentence Modelling," 373, 388–403, 10.1016/j.ins.2016.08.084.

França, Reinaldo Padilha, Ana Carolina Borges Monteiro, Rangel Arthur, and Yuzo Iano (2021) "Chapter 3 - An overview of deep learning in big data, image, and signal processing in the modern digital age," in Piuri, Vincenzo, Sandeep Raj, Angelo Genovese, and Rajshree Srivastava eds. *Trends in Deep Learning Methodologies*, 63–87: Academic Press, https://doi.org/10.1016/B978-0-12-822226-3.00003-9.

Frisby, Saskia L., Ajay D. Halai, Christopher R. Cox, Matthew A. Lambon Ralph, and Timothy T. Rogers (2023) "Decoding Semantic Representations in Mind and Brain," *Trends in Cognitive Sciences*, 27 (3), 258–281, 10.1016/j.tics.2022.12.006.

Garrido, Marta I., James M. Kilner, Klaas E. Stephan, and Karl J. Friston

(2009) “The Mismatch Negativity: A Review of Underlying Mechanisms,” 120 (3), 453–463, 10.1016/j.clinph.2008.11.029.

Gori, Marco, Alessandro Betti, and Stefano Melacci (2023) *Machine Learning: A Constraint-Based Approach*: Morgan Kaufmann, 2nd edition, https://www.sciencedirect.com/science/book/9780323898591.

Han, Su-Hyun, Ko Woon Kim, SangYun Kim, and Young Chul Youn (2018) “Artificial Neural Network: Understanding the Basic Concepts without Mathematics,” 17 (3), 83–89, 10.12779/dnd.2018.17.3.83.

Ioffe, Sergey and Christian Szegedy (2015) “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 10.48550/arXiv.1502.03167.

Issa, Mohamed F., Izhar Khan, Manuela Ruzzoli, Nicola Molinaro, and Mikel Lizarazu (2024) “On the speech envelope in the cortical tracking of speech,” *NeuroImage*, 297, 120675, https://doi.org/10.1016/j.neuroimage.2024.120675.

Jeddi, Zahra, Afsaneh Doosti, Ali Hajimohammadi, and Abdolrahim Asadollahi (2025) “Association among Extended High-Frequency Hearing, Speech Perception in Noise, and Auditory Temporal-Spectral Processing in Adults with Normal Hearing: A Cross-Sectional Study,” 77 (6), 2318–2325, 10.1007/s12070-025-05496-3.

Jolliffe, Ian T. and Jorge Cadima (2016) “Principal Component Analysis: A Review and Recent Developments,” *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 374 (2065), 20150202, 10.1098/rsta.2015.0202.

Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman (2021) “1D Convolutional Neural Networks and Applications: A Survey,” *Mechanical Systems and Signal Processing*, 151, 107398, 10.1016/j.ymssp.2020.107398.

Kraus, Nina and Karen Banai (2007) "Auditory-Processing Malleability: Focus on Language and Music," 16 (2), 105–110, 10.1111/j.1467-8721.2007.00485.x.

Le-Khac, Phuc H., Graham Healy, and Alan F. Smeaton (2020) "Contrastive Representation Learning: A Framework and Review," *IEEE Access*, 8, 193907–193934, 10.1109/ACCESS.2020.3031549.

Lippi, Vittorio and Giacomo Ceccarelli (2019) "Incremental Principal Component Analysis Exact Implementation and Continuity Corrections," in *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, 473–480, 10.5220/0007743604730480.

Luo, Nan, Xiaojing Zhong, Luxin Su, Zilin Cheng, Wenyi Ma, and Pingsheng Hao (2023) "Artificial Intelligence-Assisted Dermatology Diagnosis: From Unimodal to Multimodal," 165, 107413, 10.1016/j.compbiomed.2023.107413.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto (2015) "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 18–25.

McLoughlin, Ian, Lam Pham, Yan Song et al. (2026) "Spectrogram Features for Audio and Speech Analysis," 16 (2), 10.3390/app16020572.

McWeeny, Sean and Elizabeth S. Norton (2020) "Understanding Event-Related Potentials (ERPs) in Clinical and Basic Language and Communication Disorders Research: A Tutorial," 55 (4), 445–457, 10.1111/1460-6984.12535.

Nayak, Chetan S. and Arayamparambil C. Anilkumar (2025) "Normal EEG Waveforms," in *StatPearls*: StatPearls Publishing, `http://www.ncbi.nlm.nih.gov/books/NBK539805/`.

van den Oord, Aaron, Yazhe Li, and Oriol Vinyals (2019) "Representa-

tion Learning with Contrastive Predictive Coding," 10.48550/arXiv.1807.03748.

O'Shea, Keiron and Ryan Nash (2015) "An Introduction to Convolutional Neural Networks."

Paiva, Tiago O., Pedro R. Almeida, Fernando Ferreira-Santos, Joana B. Vieira, Celeste Silveira, Pedro L. Chaves, Fernando Barbosa, and João Marques-Teixeira (2016) "Similar Sound Intensity Dependence of the N1 and P2 Components of the Auditory ERP: Averaged and Single Trial Evidence," 127 (1), 499–508, 10.1016/j.clinph.2015.06.016.

Patel, Salil H. and Pierre N. Azzam (2005) "Characterization of N200 and P300: Selected Studies of the Event-Related Potential," 2 (4), 147–154, 10.7150/ijms.2.147.

Polich, John (2007) "Updating P300: An Integrative Theory of P3a and P3b," 118 (10), 2128–2148, 10.1016/j.clinph.2007.04.019.

Radford, A., J. W. Kim, C. Hallacy et al. (2021) "Learning Transferable Visual Models From Natural Language Supervision," *arXiv.org*.

Radford, A., J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever (2022) "Robust Speech Recognition via Large-Scale Weak Supervision."

Rugg, M.D. (2009) "Event-Related Potentials (ERPs)," in Squire, Larry R. ed. *Encyclopedia of Neuroscience*, 7–12, Oxford: Academic Press, https://doi.org/10.1016/B978-008045046-9.00752-X.

Sarker, Iqbal H. (2021) "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," 2 (6), 420, 10.1007/s42979-021-00815-1.

Shams, Siavash, Richard Antonello, Gavin Mischler, Stephan Bickel, Ashesh Mehta, and Nima Mesgarani (2025) "Neuro2Semantic: A Transfer Learning Framework for Semantic Reconstruction of Continuous Language from Human Intracranial EEG," May, 10.48550/arXiv.2506.00381.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014) "Dropout: A Simple Way to Prevent Neural Networks from Overfitting."

Sur, Shravani and Vk Sinha (2009) "Event-Related Potential: An Overview," 18 (1), 70, 10.4103/0972-6748.57865.

Taye, Mohammad Mustafa (2023) "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," 11 (3), 10.3390/computation11030052.

Touvron, Hugo, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou (2021) "Going Deeper with Image Transformers," April, 10.48550/arXiv.2103.17239.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017) "Attention Is All You Need," August, 10.48550/arXiv.1706.03762.

Walczak, Steven and Narciso Cerpa (2003) "Artificial Neural Networks," in Meyers, Robert A. ed. *Encyclopedia of Physical Science and Technology (Third Edition)*, third edition edition, 631–645, New York: Academic Press, https://doi.org/10.1016/B0-12-227410-5/00837-1.

Wilroth, Johanna, Oskar Keding, Martin A. Skoglund, Maria Sandsten, Martin Enqvist, and Emina Alickovic (2026) "Neural Tracking of Sustained Attention, Attention Switching, and Natural Conversation in Audiovisual Environments Using Wearable EEG," *European Journal of Neuroscience*, 63 (9), e70538, https://doi.org/10.1111/ejn.70538, e70538 EJN-2025-12-33601.

World Health Organization (2021) *World report on hearing*, Geneva: World Health Organization, https://www.who.int/publications/i/item/9789240020481.

Xue, Yihao, Eric Gan, Jiayi Ni, Siddharth Joshi, and Baharan Mirzasoleiman

(2024) “Investigating the Benefits of Projection Head for Representation Learning,” 10.48550/arXiv.2403.11391.

Yu, Fisher and Vladlen Koltun (2016) “Multi-Scale Context Aggregation by Dilated Convolutions,” April, 10.48550/arXiv.1511.07122.

Zhang, Qingchen, Laurence T. Yang, Zhikui Chen, and Peng Li (2018) “A Survey on Deep Learning for Big Data,” 42, 146–157, 10.1016/j.inffus.2017.10.006.

Zhou, Xin, Xiaonan Wu, Suwei Ma et al. (2025) “Characteristic Cortical Alterations in Auditory Neuropathy: An EEG Study,” 468, 109468, 10.1016/j.heares.2025.109468.

Appendix

A. Additional tables and figures

Table A1. Hyperparameters used for training Model 1.

Hyperparameter	**Value**
Window length	3 s
Stride	3 s
Batch size	256
Epochs	25
Optimizer	AdamW
Learning rate	0,0003
Weight decay	0,01
Dropout	0.4 (optimal)
Target dimensions (PCA)	64 (optimal)
Hidden channels	320
Kernel size	3
Dilation rates	1, 2, 4, 8, 16
Loss function	CLIP (Temperature: 0.1)

Table A2. Hyperparameters used for training Model 2.

Hyperparameter	**Value**
Window length	3 s
Stride	3 s
Actual batch size	512
Epochs	50
Optimizer	AdamW
Learning rate	0,00003 (20% warm-up)
Weight decay	0,15
Dropout	
(Residual layers, Transformer, ProjectionHead)	(0.1, 0.2, 0.3)
Target dimensions (PCA)	~ 160, 1024 (optimal)
Loss function	InfoNCE (Temperature: 0.07)