Speaker Generalization Using Autoencoders for Reconstructing Word Articulations

Master's Thesis
July 12, 2024

Mercylyn Wiemer
6626440, m.d.wiemer@students.uu.nl

Daily Supervisors:
Dr. Julia Berezutskaya, Dr. Zachary Freudenburg

First supervisor:
Dr. Chris Klink

Second supervisor:
Dr. Mathijs Raemaekers

Master Artificial Intelligence
Utrecht University

Abstract

Individuals with neurological conditions, including brainstem stroke or progressive Amyotrophic Lateral Sclerosis (ALS), often experience severe speech and motor impairment. In some cases, this results in a complete loss of the ability to speak, as observed in locked-in syndrome (LIS). To restore communication abilities for people with LIS, assistive tools such as brain-computer interfaces (BCIs), can provide a form of communication. By using signals directly from the brain, these technologies can serve as a vital communication channel. Direct word decoding can provide a more natural way of communication by recording brain activity during attempted speech. The current study investigated speaker generalization using real-time Magnetic Resonance Imaging (rtMRI) data capturing speech dynamics of the vocal tract. We trained an autoencoder model to generate compact representations of rtMRI videos containing individual words from multiple speakers. Instead of focusing solely on data reconstruction, the compact representations were also designed to encode phoneme information of the corresponding words. Additionally, we applied a custom loss function to calculate the phonemic distance, adapted from the Levenshtein distance. We compared two types of models: the speaker-invariant model, which was trained on data from all speakers, and the speaker-specific models, which were trained on data from each individual speaker separately. The results of this study showed that the speaker-invariant model reduced the total loss (reconstruction and phoneme loss) by a factor of approximately 10 compared to the speaker-specific models, accurately reconstructing the data and effectively encoding phoneme information. Analysis of the compact representations by calculating the Euclidean distance between vectors and comparing these distances for each model revealed significant positive correlations. This suggests similar processing of the word articulations. Another finding was the impact of data quantity, with weaker correlations between speaker-specific and speaker-invariant models when participants had less data available. Future research should investigate the relationship between neural representations and the compact representations of generalized word articulations to better understand the connection between articulation patterns and neural activity.

Keywords: brain-computer interface, speech production, autoencoder, speaker generalization

Acknowledgements

This master's thesis project was conducted at the Utrecht-BCI Lab at the UMC Utrecht Brain Center. I would like to thank the Utrecht-BCI Lab. Being part of the research group has been both inspiring and motivating throughout my thesis work. Many researchers in the group supervise students from various disciplines centered around neuroscience, providing many opportunities for feedback during monthly student updates, which is greatly appreciated.

I would like to express my gratitude to my daily supervisor Dr. Julia Berezutskaya, for her guidance and feedback during the project. Additionally, I want to thank Dr. Zachary Freudenburg, who supervised the first half of my thesis project during Dr. Berezutskaya's maternity leave. I also want to thank my university supervisor Dr. Chris Klink, for his feedback and reassuring perspective when research challenges arose.

I am deeply thankful to my parents for their support throughout my studies. Last, but not least, I want to thank my partner for being there for me during my moments of hesitation, as well as during the uplifting times.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context

Communication is one of the most important characteristics of humans, allowing us to share thoughts and express emotions. Humans engage in daily interactions through spoken or sign language, gestures, and facial expressions. Difficulties with speaking can have a significant impact on one's quality of life, especially for those who lose their ability to communicate effectively. Patients with vocal fold paralysis, a condition that severely impairs speaking, frequently report experiencing social isolation and frustration due to the limitations in communication (Francis *et al.*, 2018). Neurological conditions, such as Parkinson's Disease (PD), have a high incidence of speech disorders, with estimates indicating that up to 89% of individuals with PD are affected (Trail *et al.*, 2005). Communication impairments in PD are caused by both motor and cognitive dysfunction, as speech production requires the integration of motor and cognitive processes in real time (Smith & Caplan, 2018).

A similar pattern is observed in Amyotrophic Lateral Sclerosis (ALS), a neurodegenerative disorder that primarily affects the motor system (Masrori & Van Damme, 2020). Speech production in ALS is often affected by two conditions: dysarthria (difficulty in articulating speech) and dysphagia (difficulty in swallowing), as the muscles involved in swallowing, such as the tongue, are also used for speech (Ruoppolo *et al.*, 2013). These conditions can severely reduce a person's ability to speak, potentially leading to a complete inability to communicate (Ceslis *et al.*, 2020). Surveys such as that conducted by Felgoise *et al.* (2016) have shown that for individuals with ALS, impairments in verbal communication reduced the quality of life. Both ALS and brainstem strokes can result in the loss of voluntary muscle control and even cause individuals to become locked-in. In this condition, known as locked-in syndrome (LIS), individuals may retain only minimal muscle control, which severely limits their ability to communicate independently (Sellers *et al.*, 2014). LIS is a neurological condition characterized by paralysis of all four limbs and torso, along with a complete loss of speech, while preserving consciousness (Lulé *et al.*, 2009). There are different types of LIS depending on the degree of immobility. In classical LIS, individuals often retain control over vertical eye movements, which become crucial for communication. Alternative communication methods involve eye blinks or movements to indicate yes-no responses and to select letters or symbols on communication boards (Rousseau *et al.*, 2015). These methods can be slow and require significant effort. In complete LIS, individuals are unable to communicate due to total immobility (Halan *et al.*, 2021) (Smith & Delargy, 2005).

Assistive brain-computer interface (BCI) technology can enable communication for people living with paralysis (He *et al.*, 2020). A BCI is a system designed to record signals from the brain, decode the signals, and use them to operate a computer, without relying on muscle control. These brain signals can be captured in different ways. Two examples include techniques for capturing signals from the scalp using electroencephalography (EEG) and from the cortical surface using electrocorticography (ECoG) (Värbu *et al.*, 2022) (Schalk & Leuthardt, 2011). A more specific subtype, a speech brain-computer interface (BCI), produces speech output, including words, sentences, or synthesized speech, using the recorded brain signals (Rabbani *et al.*, 2019). The most studied BCI application is the BCI-speller, which frequently relies on EEG signal features (Rezeika *et al.*, 2018). A BCI-speller based on EEG data is noninvasive and improves autonomy, although the letter-by-letter communication process is slow. A growing body of research focuses on developing BCIs by investigating the brain regions involved in speech production. Decoding entire words from brain activity could offer a more efficient approach and enable more natural communication. Recent studies in BCI research integrated word decoding with artificial neural networks, effectively demonstrating the decoding of attempted speech from neural activity in individuals with ALS and with a brainstem stroke (Metzger *et al.*, 2023) (Willett *et al.*, 2023).

1.2 Problem Definition

In recent years, the domain of BCI research has gained significant attention and has demonstrated promising results to improve human lives. Advancing our understanding of speech production can benefit the development of new and advanced BCI applications. Speech production requires fast and precise movements of the vocal tract articulators (lips, tongue, and jaw) in coordination with the larynx (voice box) and the respiratory system (Conant *et al.*, 2018). Approximately 100 individual muscles are involved in natural speech production (Simonyan & Horwitz, 2011), and the direct link between these movements and speech segments (e.g. words) or even acoustic signal is highly complex. There are many techniques to record the articulatory movements of speech production (Toutios *et al.*, 2019). Over the past two decades, real-time Magnetic Resonance Imaging (rtMRI) has become a significant tool for investigating speech production, effectively capturing the movements within the vocal tract area during speech (Narayanan *et al.*, 2014). Compared to other techniques such as x-ray and electromagnetic articulography (EMA), rtMRI offers the advantages of being non-invasive and free from ionizing radiation, while still providing comprehensive dynamic imaging of the midsagittal plane of the vocal tract.

Various studies have employed rtMRI to investigate various aspects of speech, including speech synthesis (Toutios *et al.*, 2016), articulatory-to-acoustic mapping (Yu *et al.*, 2021), phoneme classification (Van Leeuwen *et al.*, 2019), and segmentation of the vocal tract and articulators (Ruthven *et al.*, 2021). A previous master's thesis project at the Utrecht-BCI Lab, where the current thesis is also being conducted, demonstrated the potential of using autoencoders, a specific type of neural network, to compress rtMRI data of speech production (Stolwijk, 2022). Their focus was on the reconstruction capacity and phoneme information captured by the learned representations of the autoencoder model. The subsequent clustering of these vectors revealed 20 distinct word articulation patterns, demonstrating the model's ability to differentiate between various word articulations. However, this study was limited by its use of data from a single speaker, which restricts the generalizability of its findings. Speaker-specific models are often used, as shown in the studies by Toutios *et al.* (2016) and Yu *et al.* (2021), due to the morphological differences in speech articulators among different speakers. However, a speaker-invariant model could potentially learn shared representations across speakers. Generalizing articulatory information aims to benefit a wider range of users by focusing on shared information that can be applied across different speakers, regardless of individual speech characteristics. These generalized articulation patterns can contribute to the development of BCIs that decode speech from brain activity in combination with deep learning. Generalized articulation patterns could potentially improve neural networks used for word decoding from attempted speech brain signals.

1.3 Research Questions

Building upon the foundation set by Stolwijk (2022), the primary objective of this thesis is to advance our understanding of articulation patterns by incorporating data from multiple speakers. By applying advanced deep learning techniques, specifically autoencoders, to high-dimensional rtMRI data, we aim to generate more insightful and dense representations that capture essential features in a compressed format. Autoencoders encode the input data into a bottleneck, which serves as the compressed representation of the data. This multi-speaker approach seeks to explore the variability and complexity introduced by different speakers, thereby investigating the generalization of word articulation patterns. The study addresses the following research questions:

> **Research Question 1** *Do speaker-invariant models improve the reconstruction ability and phoneme encoding of rtMRI speech data using autoencoders compared to speaker-specific models?*

> **Research Question 2** *How do the bottleneck representations of word articulations differ between speaker-specific models and speaker-invariant models?*

To address these questions, we will use the rtMRI videos from the publicly available USC-TIMIT speech database (Narayanan *et al.*, 2014). The videos consist of midsagittal frames of the vocal tract from 10 American English speakers articulating phonetically balanced sentences. We will apply the autoencoder architecture developed by Stolwijk (2022), which combines three-dimensional convolutions and recurrent neural networks, to reduce the dimensionality of the rtMRI videos. Two types of autoencoder models will be designed: a speaker-invariant model and a speaker-specific model. The speaker-invariant model will be trained on data from all speakers collectively, while the speaker-specific models will be trained on data from each individual speaker separately.

This study serves as a preliminary effort to demonstrate the feasibility of using rtMRI and deep learning to map articulation patterns, potentially enabling future research to link these patterns to brain activity and contribute to advancements in BCI research. By exploring the variability and complexity introduced by multiple speakers, this research aims to enhance the generalizability of autoencoder models, paving the way for more effective speech production analysis and applications in BCI research.

1.4 Thesis Outline

The structure of this thesis is as follows: First, in Section 2, we describe insights from other studies that this thesis builds upon. Then, in Section 3, we discuss the methodology, including data description, preprocessing steps, model architecture, and experimental setup. In Section 4, the results are presented. Section 5 interprets the results, answers the research questions, compares the current study to related work, discusses the limitations, and outlines future work. Finally, Section 6, the Conclusion, summarizes the thesis findings.

2 Related Work

In this section, we provide a review of relevant literature to support our research. This background knowledge includes studies on BCI technology, advanced techniques for recording speech production, and analyzing rtMRI data with deep learning. Subsequently, we will explore the findings from a previous project conducted at the same Utrecht-BCI lab, which this thesis builds upon.

2.1 Brain-computer Interface

Given that movements are decoded in the motor cortex, which plays a key role in coordinating voluntary muscles, much of BCI research is centered around this region. We will begin by discussing the significance of BCI technology for individuals with Locked-In Syndrome (LIS), followed by an exploration of several motor-based BCI studies and applications.

LIS is a rare neurological condition, mentioned before in Section 1.1, characterized by motor paralysis, which can result in the inability to speak (Bruno *et al.*, 2009). Limited communication sometimes is possible through eye movements, such as answering closed questions by blinking. The use of BCI technology can further assist in communication. In a study by Vansteensel *et al.* (2016), a method for communication in locked-in individuals with late-stage ALS was described, involving the control of a computer typing program based on attempted hand movements. A recent study conducted by Moses *et al.* (2021) integrated BCI technology with deep learning techniques to decode attempted speech from recorded cortical activity of the sensorimotor cortex in individual with anarthria, the loss of speech. Direct word decoding offers a more natural an faster form of communication. While the primary focus of communication restoration is on speech output (e.g., words), another important goal is to restore facial movements related to speaking. Metzger *et al.* (2023) developed a facial-avatar animation for controlling facial gestures. Animating a facial avatar to accompany synthesized speech can lead to more natural communication. This was achieved by decoding articulatory and orofacial representations from the speech-motor cortex.

2.2 Techniques for Capturing Speech

Various measurement techniques are available to capture the movements of (parts of) the vocal tract during speech production. Ultrasound imaging utilizes sound waves to capture real time images of the whole tongue, spanning from the tip to the root (Wilson, 2014). This technique is especially suited for recording the shapes and movements of the tongue during speech, making it particularly well-suited for tongue shape analysis (Dawson *et al.*, 2016). Another technique, known as electromagnetic articulography (EMA), utilizes alternating electromagnetic fields to record the real-time movements of speech articulators, including the tongue, lips and jaw. Sensors are strategically placed on these articulators for precise data capture (Katz *et al.*, 1999) (Rebernik *et al.*, 2021). Magnetic Resonance Imaging (MRI) produces detailed images of internal structures, applying large magnets to create a strong magnetic field. Protons in the body align with this field, and radio frequency pulses disturb their alignment. When the pulses cease, the returning signals from aligned protons are detected and used to construct an image (Berger, 2002). Real-time Magnetic Resonance Imaging (rtMRI) directly acquires moving image data in contrast to the term dynamic MRI, which relates to the source, such as creating images from an actively articulating subject rather than a static postural source. This distinction underscores the emphasis on acquisition of dynamic movements in real time (Narayanan *et al.*, 2004). In speech production research, rtMRI offers dynamic insights from the complete midsagittal plane of a speaker's upper airway, or other planes of interest, providing continuous utterances without the need for repetitions. The midsagittal rtMRI allows for capturing the motion of the vocal tract during speech, encompassing the velar and pharyngeal regions. The velar region is located near the soft part of the roof of the mouth, known as the soft palate, while the pharyngeal region is located in the pharynx, the cavity behind the nose and mouth leading to the larynx. These areas are not captured by EMA (Toutios & Narayanan, 2016) (Kim *et al.*, 2014).

2.3 Applying Deep Learning to rtMRI Data

A substantial body of literature explores speech production using deep learning, with studies relying on midsagittal rtMRI data of the vocal tract area. This technique is particularly well-suited for studying the dynamic aspects of speech, benefiting from its capacity for continuous image acquisition. The USC-TIMIT dataset is a popular and freely available multi-speaker rtMRI speech database (Narayanan *et al.*, 2014). A detailed description of this dataset is provided in Section 3.1.

Deep learning has significantly reshaped various domains, for example, computer vision, language understanding and speech recognition. Over the past decade, the predominant approach to training machine learning models has been the implementation of deep neural networks (Menghani, 2023). Deep learning employs multi-layered computational models with non-linear transformations to automatically acquire increasingly abstract data representations, facilitating the learning of complex functions (LeCun *et al.*, 2015). While there are numerous deep learning architectures, most architectural designs can be adapted for a wide range of tasks, some architectures are optimized to specific data types such as time series or images. These variations are characterized by the types of layers, neural units, and connections they employ.

The study conducted by Kose and Saraclar (2021) explored multiple experiments using the USC-TIMIT dataset, extracting features from the rtMRI videos and corresponding speech data. Deep neural networks, consisting of convolutional and long short-term memory (LSTM) layers, were trained for both unimodal (audio-only or video-only) and multimodal (audio and video) approaches. These experiments covered phone classification, phone recognition, and word discrimination task. Notably, the findings revealed that employing compressed dimensional video representations not only reduced computational complexity but also enhanced the outcomes of the phone recognition task when compared to audio-only approaches. The lowest accuracy was found for the solely video input. Additionally, the study identified speaker variability as a factor contributing to errors in the word discrimination task. Another notable finding from the phone classification experiment was that most errors occurred with phones that have similar vocal tract shapes.

Another study by van Leeuwen *et al.* (2019) also investigated speech classification, specifically focusing on vowels, consonants, and phonemes in American English. They trained a convolutional neural network (CNN) to classify these speech components using rtMRI images of the vocal tract. To enhance image feature extraction and address the limited speech data, the model was pretrained on the CIFAR-10 dataset (Krizhevsky, Hinton *et al.*, 2009), which consists of 60,000 images. Additionally, data augmentation techniques such as zoom, rotation, and shift were employed to further increase the dataset. Vowel classification achieved the highest accuracy of 70.7%, consonant classification reached 61.7%, and phoneme classification was just above chance level with an accuracy of 57%.

A great deal of previous research in speech production has focused on articulatory-to-acoustic mapping. This technique is used to predict acoustic signals from speech movements, using data acquired through methods such as rtMRI or standard video. The study by Csapó (2020) demonstrated the potential of using rtMRI from the USC-TIMIT speech database for this purpose. They utilized data from four speakers and trained various deep neural networks: fully connected, convolutional neural network (CNN), and RNN. Their findings showed that combining a convolutional neural network (CNN) with LSTM units was more effective for processing rtMRI images than using a convolutional neural network (CNN) alone. Their methods included a speaker-specific approach, training a separate model for each participant, using data consisting of full sentences.

Similar to Csapó (2020), the study by Yu *et al.* (2021) employed speaker-specific models to account for the anatomical differences between speakers. They used deep neural networks, combining convolutional neural network (CNN), and RNN layers, to reconstruct speech signals from rtMRI images, training separate models for each speaker. The study evaluated the performance using Mean Absolute Error, and found large differences between speakers. The output of the networks consisted of spectral vectors, which were reconstructed into speech signals.

2.4 Speaker-independent Approach

While rtMRI studies focus on speaker-specific models due to the detailed anatomical different between individuals, speaker-independent approaches are crucial for applications like speech recognition and text-to-speech synthesis, where generalizability is essential. Research focused on improving these applications often favors speaker-independent approaches to ensure effective user interaction.

Parrot *et al.* (2020) investigated the reconstruction of articulatory trajectories from acoustic signals, with a primary focus on achieving speaker independence. They found that the speaker-independent condition, where one speaker is held out during training, resulted in lower reconstruction accuracy compared to the speaker-specific setting. This highlights the challenge of maintaining high accuracy in speaker-independent models. The ABX phone discrimination task, which evaluates a model's ability to distinguish between different phonetic units, provided additional insights. This evaluation method showed that the speaker-independent model not only retained linguistically relevant information but also improved the reconstruction of the articulatory information. This improvement was not evident through reconstruction accuracy alone, highlighting the value of the ABX phone discrimination measure for assessing model performance.

The process of voice conversion, where the voice of a speaker is transformed to sound like another speaker, is especially interesting for personalized text-to-speech systems. This is another example where speaker-independence is of importance. Mohammadi and Kain (2014) demonstrated this by training an autoencoder model on multiple speaker (11 participants) to create a speaker-independent model for compressed representations of speech spectral features. This method significantly improved the ability to convert voices across different speakers, highlighting the effectiveness of using a speaker-independent autoencoder model for voice conversion purposes.

2.5 Preceding Study Insights

As briefly mentioned in Section 1.2, the master's thesis project by Stolwijk (2022) forms the foundation for the current study, both conducted in collaboration with the Utrecht-BCI Lab. In this section, we will describe the context of the previous project and highlight its important and relevant findings, in addition to what was already described in the previous section.

The aim of the study was to identify 20 words that had the most distinct articulation patterns. These patterns were extracted from midsagittal rtMRI videos from the USC-TIMIT (Narayanan *et al.*, 2014) speech database. To effectively cluster the words, the study reduced the dimensionality of the data using two autoencoder architectures. The first architecture, Three-dimensional Convolutional Neural Network (3D-CNN), employed three-dimensional convolutions, while the second, Convolutional Gated Recurrent Unit (ConvGRU), combined three-dimensional convolutions with recurrent neural networks. After reducing each word to a representative vector, the vectors are clustered into 20 groups. The representatives of these clusters are presented as the 20 most distinct words.

Reviewing the autoencoder architectures, the main difference lies in the layers used: recurrent layers work well with sequential data and do not require padding, while 3D-CNN requires a fixed input size, necessitating padding due to the varying lengths of the data. Additionally, these models were trained on data from a single participant. This approach was chosen due to anatomical differences between participants, a factor highlighted in previous research that also employed speaker-specific methods (Csapó, 2020) (Yu *et al.*, 2021).

Furthermore, to incorporate linguistic information into the model, the corresponding phonemes of each word were one-hot encoded. This one-hot encoding was provided to autoencoders. To measure the differences between word pronunciations, the Levenshtein distance was adapted to the Phonemic Levenshtein Distance (PLD), measuring the distance based on phonemes. The custom loss function included the reconstruction loss (MSE) and Phonemic Levenshtein Distance (PLD). The latter minimized the phonemic distance between word articulations during training. A more detailed explanation is described in Section 3.4.4.

In evaluating the performance of the autoencoder architectures, the ConvGRU model, which combines convolutional and recurrent layers, demonstrated lower reconstruction loss compared to the 3D-CNN model. One possible explanation for this difference is that the 3D-CNN model required padding due to the varying lengths of the data, and this padding comes with a cost. Another experiment focused on cross-participant transferability by training the model on participant F1 and testing it on other participants. Fine-tuning the model by adding data from the unseen participant improved the reconstruction loss, with the lowest loss observed when adding 500 data samples. Overall, the ConvGRU model demonstrated better reconstruction performance and generalizability across participants.

3 Methodology

As previously mentioned, this thesis builds on the work by Stolwijk (2022), which applied a convolutional recurrent autoencoder architecture to compress high-dimensional rtMRI data of word articulations. However, unlike the previous study that focused on data from a single participant, we trained our model using data from ten speakers. This section discusses the methods used to address the research questions, with a main focus on the performance differences between speaker-specific models and a speaker-invariant model. The following subsections provide details of the data, explain the preprocessing steps, describe the network architecture and settings, outline the experimental setup, and present the evaluation methods.

3.1 Data Description

The publicly available USC-TIMIT speech production database (Narayanan *et al.*, 2014) from the University of Southern California was used in this work. The dataset contains rtMRI recordings from ten speakers (five female and five male) of American English, providing 1.5-T images of the midsagittal plane of the vocal tract. The images have a resolution of 68 × 68 pixels and a frame rate of 23.18 frames per second. Figure 1 shows a single rtMRI frame of each participant, highlighting the variability between speakers (Toutios & Narayanan, 2016). Simultaneous audio recordings were also collected, featuring 460 sentences from the MOCHA-TIMIT database (Wrench, 2000). The MOCHA-TIMIT corpus features phonetically balanced sentences, as it was originally designed to record EMA data for training an automatic speech recognition system. Notably, these sentences were written in British English, whereas the participants in the USC-TIMIT dataset spoke American English. In Table 3.1, three sentences are illustrated with examples of British English spellings. The dataset also includes transcription files that provide detailed information about the start and end times when each sentence was spoken, as well as the individual words and phonemes within those sentences. In our experiments, we used the rtMRI video data without audio from all ten participants.

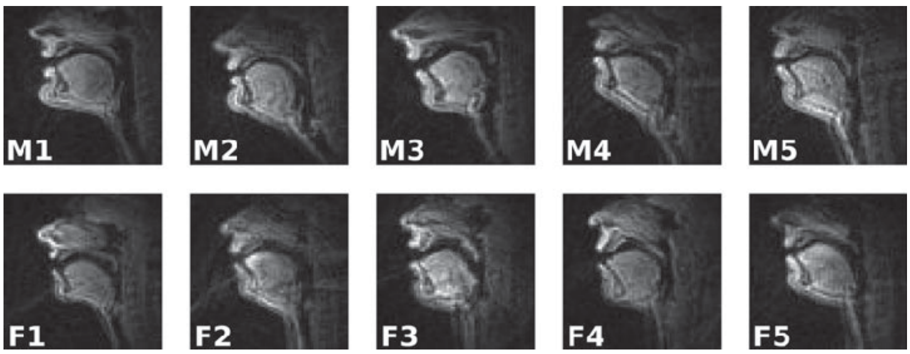


Figure 1: Example frames from the rtMRI videos of the USC-TIMIT database, including ten speakers: 5 male (top row) and 5 female (bottom row). Figure from Toutios and Narayanan (2016).

Example Sentences	American English Spelling
411: Those musicians harmonize **marvellously**.	marvelously
433: I **honour** my mum.	honor
438: We apply auditory **modelling** to computer speech recognition.	modeling

Table 3.1: Three sentences from the MOCHA-TIMIT corpus (Wrench, 2000), indicated by their order number. Words in bold are examples of British English spellings, with their American English spellings provided in the second column.

3.2 Data Preprocessing

The preprocessing pipeline included data segmentation, frame processing, and data filtering. These steps were crucial for preparing the data to be suitable as input into a neural network. Another important step was adding phoneme encodings to the rtMRI videos to correspond to word articulation.

3.2.1 Data Segmentation

The original dataset consisted of 460 sentences per participant, with each video containing five sentences. These videos were segmented into individual words based on transcription files. The segmentation was done by selecting the frames corresponding to the start and end times of each word as indicated in the transcription files. Although this preprocessing step was performed by a colleague, it is important to note that the data available for each participant varied. See Table 3.3 in Section 3.2.3 for the number of data points per participant. Participants F4 and M5 had approximately 1,000 fewer data points due to missing frames in the videos before segmentation. Participant F4 had missing frames in 35 videos, resulting in 175 fewer sentences, while participant M5 had missing frames in 24 videos, resulting in 120 fewer sentences. Since the length of these sentences varies, the number of missing words also varies.

3.2.2 Frame Processing

To prepare the data for the model experiments, several processing steps were applied to the video frames. Although the videos were in black and white, the frames were stored with three RGB color channels, which did not provide additional information. Therefore, we converted the frames from RGB to grayscale, reducing the three color channels (red, green, and blue) to a single intensity channel. To further reduce the model complexity, we applied pixel normalization by rescaling the pixel values from the range 0-255 to 0-1. High numbers increase computational complexity, so normalization makes computation more efficient by allowing the neural network to process smaller, more manageable values.

In Figure 1, it can be observed that the frames contain many black pixels outside the vocal tract area. These pixels do not provide useful information regarding word articulation. Therefore, each video frame was cropped from its original resolution of 68×68 pixels to smaller dimensions of 47×47 pixels. Following the methods described by Stolwijk (2022), a standard frame size across participants was calculated. This process involved creating pixel-variance heat maps of the data per participant. The border positions (top, bottom, left, and right) were then calculated to include all pixels with above-average variance. The top and left border positions were used as anchors for cropping, reducing the number of pixels per frame from 4624 to 2209 by excluding non-informative pixels. Figure 2 shows an example of frame cropping using a video frame from participant F1.

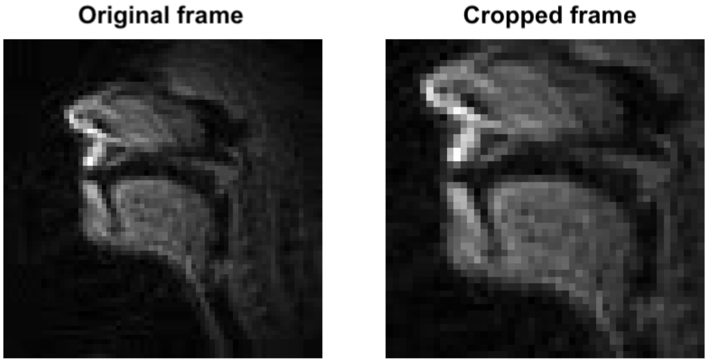


Figure 2: An example of frame cropping: Left the original frame (68 x 68 pixels). Right: the cropped frame (47 x 47 pixels), maintaining the informative pixels for word articulation.

3.2.3 Data Filtering

The words spoken in the rtMRI videos varied in length, ranging from 1 to 15 characters. The time required to pronounce these words depends not only on their length but also on the speech tempo of the participants. The number of frames in the videos corresponds to the duration of the spoken words, with longer words or slower speech tempos resulting in more frames. Figure 3 shows the frame distribution of the videos available for participant F1. The frame distribution for other participants was similar, with higher outliers for participants F3, F4, F5, and M4. The highest frame count, 99 frames, was found in the data of participant F5. See Appendix A for the frame distribution histograms of the other participants. Also, considering the video rate of 23.18 frames/sec, videos with a low number of frames may not contain sufficient information to represent the word articulation. In accordance with Stolwijk (2022), video data was selected with a minimum of 5 frames. We considered frames above 35 to be incorrectly processed during segmentation, given that the longest words consist of 15 characters.

For each video, the phonemes of the corresponding word label were extracted. We used the North American English CMU Pronouncing Dictionary (CMUDict) (Carnegie Mellon University, 1998), accessed via the Natural Language Tool-Kit (NLTK) library (Bird *et al.*, 2009). There are a total of 39 phonemes in the CMUDict corpus, as shown in Table 3.2. To store the phoneme information, we applied a one-hot encoding representation of 15 by 39, representing the maximum number of phonemes and the number of phonemes in the CMUDict corpus, respectively. Because the original words were spelled in British English, a small set of words was not recognized by the CMUDict. Table 3.1 provides three spelling examples. To extract the phoneme information, we first needed to convert these words to American English spelling.

Finally, as previously mentioned, we filtered the data to include only videos with a frame count between 5 and 35. Additionally, the words in these videos needed to be present in the CMUDict corpus. Table 3.3 presents the number of videos after data filtering, with a total of 21,777 videos including data from all participants.

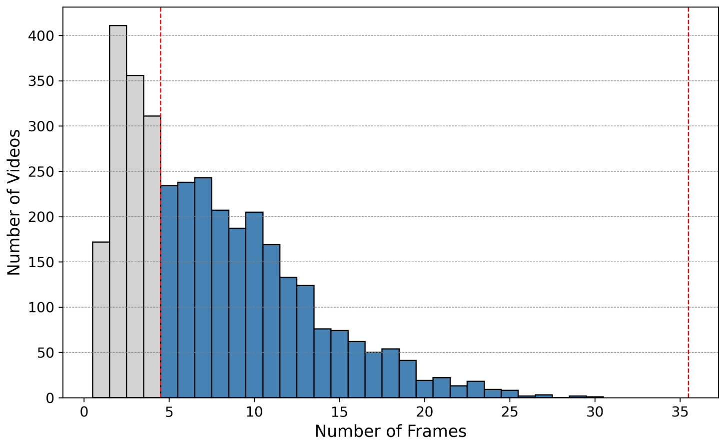


Figure 3: Histogram showing the distribution of frame counts for participant F1. Bars highlighted in blue and positioned between the red lines represent the number of videos with frame counts between 5 and 35.

Phoneme	**Example**	**Translation**	**Phoneme**	**Example**	**Translation**
AA	odd	AA D	L	lee	L IY
AE	at	AE T	M	me	M IY
AH	hut	HH AH T	N	knee	N IY
AO	ought	AO T	NG	ping	P IH NG
AW	cow	K AW	OW	oat	OW T
AY	hide	HH AY D	OY	toy	T OY
B	be	B IY	P	pee	P IY
CH	cheese	CH IY Z	R	read	R IY D
D	dee	D IY	S	sea	S IY
DH	thee	DH IY	SH	she	SH IY
EH	Ed	EH D	T	tea	T IY
ER	hurt	HH ER T	TH	theta	TH EY T AH
EY	ate	EY T	UH	hood	HH UH D
F	fee	F IY	UW	two	T UW
G	green	G R IY N	V	vee	V IY
HH	he	HH IY	W	we	W IY
IH	it	IH T	Y	yield	Y IY L D
IY	eat	IY T	Z	zee	Z IY
JH	gee	JH IY	ZH	seizure	S IY ZH ER
K	key	K IY			

Table 3.2: Phoneme set, table adapted from the CMU Pronouncing Dictionary (Carnegie Mellon University, 1998).

Participant	Number of videos
F1	2181
F2	2410
F3	2319
F4	1358
F5	2362
M1	2494
M2	2455
M3	2204
M4	2518
M5	1478
Total	21 777

Table 3.3: Number of rtMRI videos per participant in the dataset after preprocessing: word labels recognized by the pronouncing dictionary and with frame counts between 5 and 35.

3.3 Model Architecture

Following the approach by Stolwijk (2022), we employed the Convolutional Gated Recurrent Unit (ConvGRU) autoencoder architecture due to its low reconstruction loss and greater generalizability across participants compared to the completely convolutional architecture. Since we aim to investigate word articulations in video data, the ConvGRU is well suited for this task. It combines convolutional and recurrent layers, effectively leveraging their strengths for processing sequential image data.

3.3.1 Autoencoders

An autoencoder is a type of neural network used in unsupervised learning, designed to learn efficient data representations. Figure 4 shows a simple autoencoder architecture. The architecture consists of two main parts: an encoder, which compresses input into a low-dimensional representation (or bottleneck), and the decoder, which reconstructs the original input from this bottleneck. Within the network's internal structure, the hidden layer h encodes the bottleneck representation using the encoder function, $h = f(x)$, and the decoder reconstructs the input using the function, $r = g(h)$ (Goodfellow *et al.*, 2016).

3.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs), introduced by LeCun *et al.* (1998), are specialized neural networks for processing grid-like data structures such as images (Goodfellow *et al.*, 2016). Unlike, fully connected neural networks, CNNs retain the spatial information of the input data. Instead of connecting every single neuron to the next layer, CNNs connect subsections of the input, known as patches, to the next layer. This approach leverages the fact that pixels that are close to each other are more likely to be similar than those farther apart. The main mathematical operation of CNNs, is the convolution, which involves element-wise multiplication of a specific filter (kernel), that moves across the input image. This process generates a feature map that includes specific features of the previous layer.

For image data, two-dimensional convolutions are appropriate because they handle the spatial dimensions of width and height. However, video data contains a third dimension: temporal information across multiple frames. Three-dimensional convolutions extend two-dimensional convolutions by adding this extra temporal dimension to capture both spatial and temporal features (Ji *et al.*, 2013).

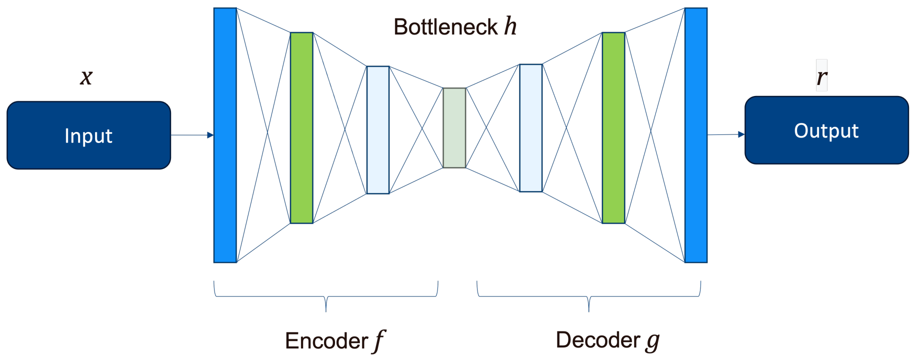


Figure 4: Simplified autoencoder architecture with the following components: the encoder f, the bottleneck h and the decoder g. The input is represented by x, and the output (reconstruction) is represented by r.

3.3.3 Recurrent Neural Network

In a neural network, the hidden layer transforms input data by computing a weighted sum of its inputs and subsequently applying an activation function, creating a new representation of the input. In a recurrent neural network (RNN), the hidden layer forms a cycle, allowing the network to retain memory of past inputs. This cyclic structure enables the hidden layer's activation to depend not only on the current input but also on its activation from the previous time step. Memory retention is crucial for tasks involving sequential data (Jurafsky & Martin, 2024).

RNNs are trained through backpropagation. However, they often encounter two challenges: the vanishing gradient problem, where gradients become extremely small, and the exploding gradient problem, where gradients become too large. These problems can make it challenging for the network to capture long-term dependencies in sequential data (Bengio *et al.*, 1994). A variant of the RNN, known as the Gated Recurrent Unit (GRU) has been introduced to address these problems (Cho *et al.*, 2014). It employs a gating mechanism that enables selective updates and resets of the hidden state, providing improved control over long-term dependencies. Notably, the GRU architecture, which is less complex than the Long Short-Term Memory (LSTM) RNN variant, achieves computational efficiency while delivering robust performance in tasks requiring memory retention.

3.3.4 ConvGRU Architecture

The Convolutional Gated Recurrent Unit (ConvGRU) architecture was inspired by the convolutional autoencoder proposed by Chong and Tay (2017). As illustrated in Figure 5, the encoder consists of two three-dimensional convolutional layers and one recurrent layer, while the decoder consists of one recurrent layer and two transposed convolutional layers. The input tensor has a size of 1 x (5 x 47 x 47), where 1 is the input channel, 5 is the number of frames, and 47 x 47 is the image height and width. In the encoder, each three-dimensional convolutional layer is followed by a rectified linear unit (ReLU) activation function to introduce non-linearity. In the first convolutional layer, the input tensor is transformed to an output tensor with dimensions 128 x (5 x 23 x 23). This transformation involves applying 128 filters, each with a kernel size 1 x 3 x 3 and a stride of 1 x 2 x 2. In the second convolutional layer, this output tensor is further transformed to dimensions 32 x (5 x 11 x 11). This transformation involves applying 32 filters, using the same kernel size and stride as the previous layer.

The layers in blue represent the encoder, the rectangle in red indicates the bottleneck of vector size 100, and the layers in purple illustrate the decoder. The recurrent layers, part of both the encoder and decoder (see green arrows), handles temporal dependencies and is composed of convolutional GRU cells. The hidden states of these GRU cells are important for temporal feature learning.

Additionally, the phonemes sequence is illustrated in orange. Initially, this sequence has a shape of 15 x 39 and is flattened to a vector of size 585. The hidden state of the GRU cell, are flattened from 32 x 11 x 11 to a vector of size 3872. These two flattened vectors are concatenated to a vector of size 4,457 and passed through a linear layer, reducing its dimensionality from 4,457 to 100.

Subsequently, the 100-dimensional vector (the bottleneck) is passed through another linear layer with a Tanh activation function, expanding it to a size of 3872. The output from this linear transformation, together with the output from the first ConvGRU layer (32 x (5 x 11 x 11)), is then passed through a second ConvGRU layer. This is followed by the two three-dimensional transposed convolutional layers with ReLU activation functions, producing the reconstructed output with the same dimensions as the original input.

5

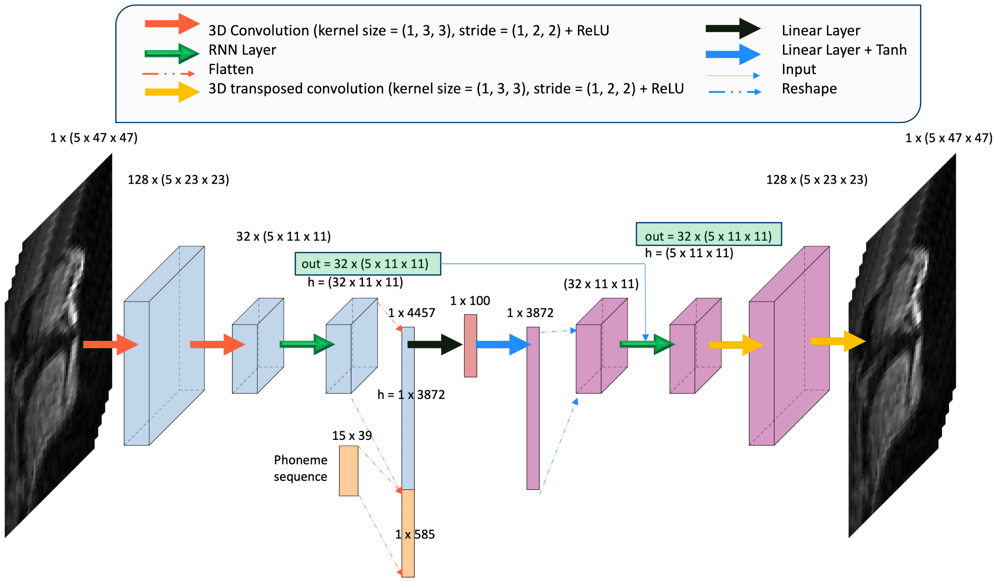


Figure 5: The architecture of the ConvGRU autoencoder. Adapted from Fig 5 in Stolwijk (2022). Blue blocks represent the encoder, processing input data through a series of layers. The red rectangle indicates the bottleneck. Purple blocks illustrate the decoder, reconstructing data from the bottleneck representation. The phoneme sequence is highlighted in orange.

3.4 Experimental Setup

In this section, we describe the experimental design, training details, phoneme information, and evaluation metrics.

3.4.1 Experimental Design

To investigate speaker generalization using autoencoders, we employed two categories of models: speaker-specific and speaker-invariant. The speaker-specific models were trained and validated on data from individual participants, while the speaker-invariant model was trained and validated on data from multiple participants. In total, we trained 10 speaker-specific models and one speaker-invariant model. To compare the results, we evaluated the speaker-invariant model using the same test set as the speaker-specific models, ensuring the test data was specific to each participant.

3.4.2 Training Details

Each model was trained for a maximum of 200 epochs. Early stopping was applied if the validation loss did not improve after 15 consecutive epochs. For training, the Adam optimizer (Kingma & Ba, 2014) started with an initial learning rate of 0.0003. A dynamic learning rate decay technique was applied using the ReduceLROnPlateau method (factor = 0.8 and patience = 10), based on the validation loss. Similar to the study by Stolwijk (2022), a weight decay of 10^{-8} was used to reduce the risk of overfitting. After the pre-processing steps, the data was randomly split into a ratio of 8:1:1, where 80% was training data, 10% was validation data, and 10% was test data. All the models were implemented in PyTorch (Paszke *et al.*, 2017), using a single NVIDIA GeForce RTX 2080 Ti GPU for training.

For all experiments, the mini-batch training method was employed with a batch size of 10. This approach was chosen due to the variability in data size, as the data consists of different frame lengths. Additionally, RNNs require the same input dimensions at each time step for proper sequence processing. By using mini-batches, the model processes batches with the same number of frames, ensuring consistent training. When a group of data points with the same frame count exceeds 10, multiple mini-batches are created. Each mini-batch contains up to 10 data points, except for the last batch, which may contain fewer than 10 data points if the total number is not a multiple of 10. This strategy ensures that all data is utilized effectively, without dropping any smaller batches.

The validation loss was used to save the best-performing model. After each epoch, the validation loss was compared with the loss of the last saved best-performing model. The model's parameters were adjusted based on the training data. The validation set helped monitor and select the best-performing model without directly updating the parameters. However, using the validation set for model selection can introduce bias, as the model may become tuned to perform well on the validation data rather than generalizing to new, unseen data. The selected model was then used to test performance on the test set and assess how well it generalized to new data. In the results section, the evaluation metrics refer to the average batch performance on the test set.

3.4.3 Phoneme information

The objective of traditional autoencoders is to minimize the reconstruction error between the original input and the reconstructed output. This forces the model to learn a compressed and meaningful representation of the input data. For the current study, we are specifically interested in the representation learning of word articulations in rtMRI videos. To improve these representations, we incorporated a second data stream containing the phonemes of the words spoken in the videos. Phonemes are the smallest units of sound that distinguish words. By combining this linguistic information with the articulation patterns observed in the rtMRI videos, the model can learn more informative features. As mentioned in Section 3.2.3, for each data point, we extracted the phonemes of the word label and encoded these phonemes using one-hot encoding.

3.4.4 Evaluation Metrics

To evaluate the performance of the models, we computed two metrics: the reconstruction loss and the Phonemic Levenshtein Distance (PLD) loss, which incorporates the phoneme information. These metrics were combined into a single total loss value during training and validation. The reconstruction loss, specifically the mean squared error (MSE) between the input videos and the reconstructed videos, was defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

with y_i representing the ground truth, $\hat{y}_i$ representing the predicted output by the autoencoder, and n being the number of data points in the batch. The loss was calculated across all elements in a batch, providing an average batch loss.

Following the methodology of Stolwijk (2022), we adopted a custom loss function inspired by the Levenshtein distance metric (Levenshtein *et al.*, 1966). This specific distance metric calculates the difference between two string sequences based on the minimum number of operations needed to convert one word to the other. There are three edit operations: insertion, deletion, or substitution, of a character. In Figure 6, we show an example of calculating the Levenshtein distance between the words ‘brain’ and ‘rain’. There is one edit needed, namely the deletion of the character ‘b’.

B	R	A	I	N	Deletion B
	R	A	I	N	

Figure 6: Levenshtein Distance Example: The difference between ‘brain’ and ‘rain’ is one edit (one deletion).

The PLD applies this concept of Levenshtein distance to phonemes. The difference between the phonemes of two words is calculated by counting the number of edit operations. Figure 7 shows an example with the words ‘pear’ and ‘pair’. Because these words consist of the same phonemes, the PLD is zero. We employed the custom loss function by first calculating the PLD between all words in a batch. Then, the Euclidean distance was computed between the generated bottleneck representations for each word. Finally, we calculated the MSE between these two distance matrices. Combining the two loss functions, we end up with the following total loss function:

$$\text{Loss} = R + wP$$

Here, R represents the reconstruction loss, and P is the PLD loss scaled by a weight w. We used a weight of 0.006, consistent with the findings of Stolwijk (2022).

Word	P	E	A	R
Phonemes	P	EH		R

Word	P	A	I	R
Phonemes	P	EH		R

Figure 7: Phonemic Levenshtein Distance Example: The difference between ‘pear’ and ‘pair’ is zero because the words consist of the same phonemes.

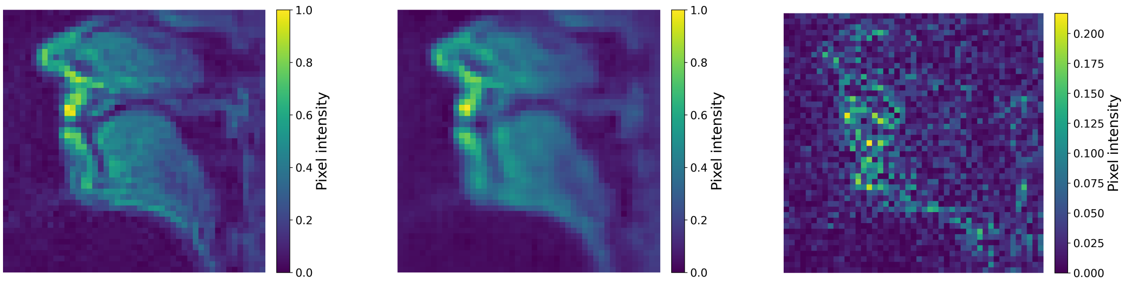
3.4.5 Model Performance Visualization

To evaluate and compare the performance of the speaker-specific and speaker-invariant models, we conducted two main analyses by visualizing the reconstructed outputs and bottleneck representations.

Reconstruction of Individual Data Points

We reconstructed individual data points from each participant in the test set and visualized one frame from each video. This visualization consisted of the original frame, the reconstructed frame, and the difference between them. This analysis was done for both the speaker-specific model and the speaker-invariant model to make the results more interpretable and to compare the reconstructions between these models. An example of this visualization is shown in Figure 8.

The colorbar of Figure 8c is different from those in Figures 8a and 8b. In this plot, the values were scaled to highlight the differences between the original frame and the reconstructed frame. Since the differences are small, the colorbar is scaled to the highest pixel value to clearly illustrate the difference. This example visualization shows the speaker-invariant model after training for one epoch.



(a) Original frame (b) Reconstructed frame (c) Difference

Figure 8: Example visualization of input reconstruction: Frames originally from a video of participant F1. The MSE of the pixel values between the original and reconstructed frame was 1.8×10^{-3}.

Similarity Matrices

Furthermore, we generated similarity matrices using the Euclidean distance between data points. The bottleneck representation, a compressed representation of a word articulation as a 100-dimensional vector (1 x 100), allows for efficient distance computation between these vectors. These distances were visualized in a heatmap, enabling a comparison between the speaker-specific model and the speaker-invariant model based on their bottleneck vectors. This comparison was performed using the test set, which contained approximately 200 data points per participant. Through pairwise comparison, around 19,900 unique comparisons were obtained per participant, excluding the diagonal which compares the same data points. Further analysis included visualizing repeated words and short words. Finally, the upper triangle of the Euclidean distance matrices, excluding the diagonal, was flattened for each participant. These vectors were then correlated between the speaker-specific and speaker-invariant models using the Spearman rank-order correlation coefficient.

4 Results

In this study, we trained speaker-specific and speaker-invariant models to reduce high-dimensional rtMRI videos of word articulations to representative vectors that encode the phonemes of the spoken words. First, we will discuss the model performance and illustrate the reconstruction ability with a sample frame from participant F1. Second, we will compare the bottleneck representations of the different models with the data from participants F1 and M5.

4.1 Reconstruction and Phoneme Loss

The models were trained for different epochs, as we implemented early stopping. Consequently, the training duration for the speaker-specific models varied, ranging from 30 minutes to 4 hours. The speaker-invariant model took around 12 hours to train. Table 4.1 shows the epoch at which the validation loss was the lowest for each model. The speaker-invariant model was trained on all data, so it has only one best epoch value.

Model	**F1**	**F2**	**F3**	**F4**	**F5**	**M1**	**M2**	**M3**	**M4**	**M5**	**Invariant**
Best Epoch	93	124	86	130	114	194	197	194	191	134	88

Table 4.1: Best epoch for speaker-specific models (indicated with the participant number), and speaker-invariant model.

As described in Section 3, we compared model performance by evaluating the models based on the MSE loss and PLD loss. Figures 9 and 10 present the average MSE and PLD loss, respectively, on the test set per participant for each model: speaker-specific and speaker-invariant. From these results, we can see that the performance of the speaker-invariant model shows lower loss values for both MSE and PLD loss. Applying the non-parametric Wilcoxon signed-rank test, we found a significant difference between the two model categories (speaker-specific and speaker-invariant) when comparing the average test loss per participant for both MSE ($p = 0.002$, Wilcoxon statistic $= 0.0$) and PLD ($p = 0.002$, Wilcoxon statistic $= 0.0$). It is also apparent from Figures 9 and 10 that participants F4 and M5 have the highest loss values, which is likely because these subject have less data available (see Table 3.3). The specific loss values plotted in Figures 9 and 10 are provided in Appendix B.

4.1.1 Visualization of reconstruction performance

To illustrate the difference in reconstruction ability between the speaker-specific and speaker-invariant models, we present sample reconstruction plots in Figures 11 and 12, showing the reconstruction of a single frame from a video in the test set. The same data point was used, where participant F1 spoke the word 'Puree'. The original frames, which are the same in both figures, are shown in Figures 11a and 12a. For this particular frame, the MSE was 0.00032 for the speaker-specific model and 0.000019 for the speaker-invariant model. The videos consisted of grayscale pixels, with pixel intensities ranging from 0 to 1. To enhance the visibility of pixel differences, we used a more distinguished colormap for plotting the frames. Additionally, because the differences in pixel values are small, the colorbars in Figures 11c and 12c are scaled from 0 to the maximum difference value, which was found in the speaker-specific model plot, ensuring a consistent colorbar. From Figures 11b and 12b, we can observe that the reconstructed frames from both models are very similar to the original frame. Figures 11c and 12c demonstrate the difference between the original frame and the reconstructed frame, showing higher pixel differences for the speaker-specific model.

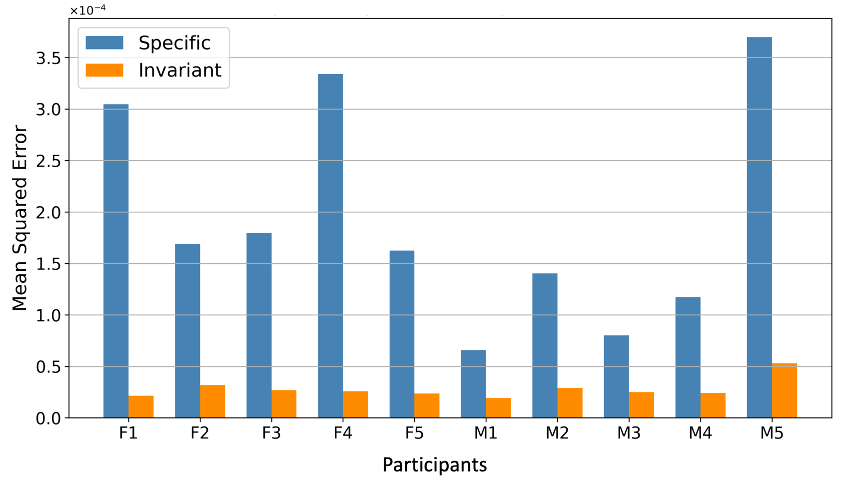


Figure 9: Average MSE loss on the test set: speaker-specific results are shown in blue, and speaker-invariant results are shown in orange.

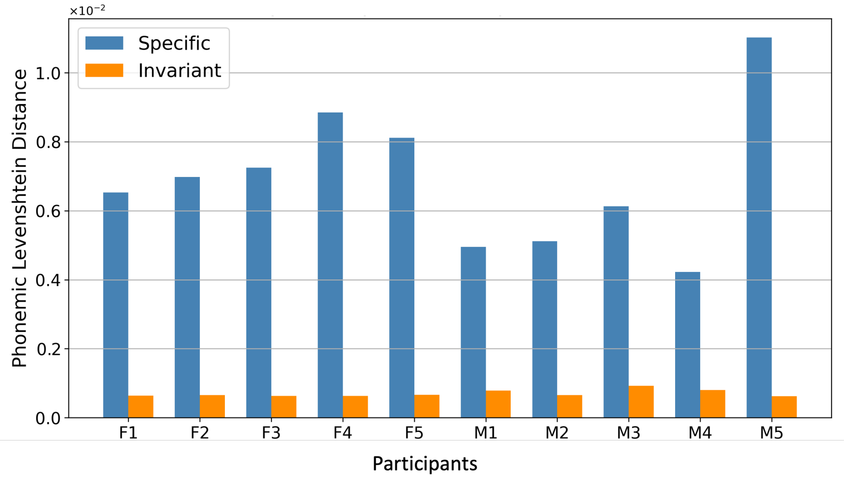
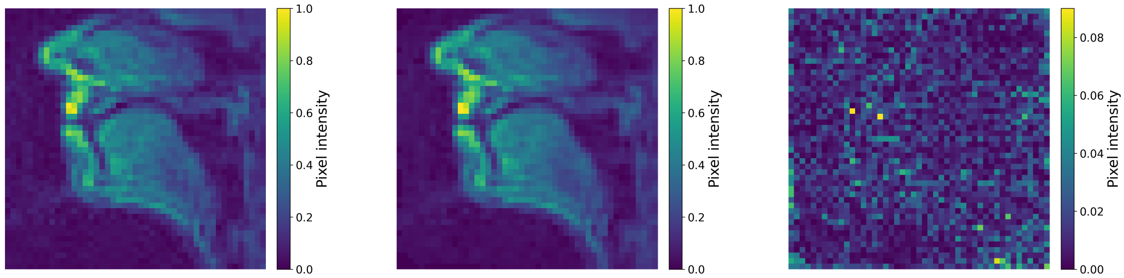
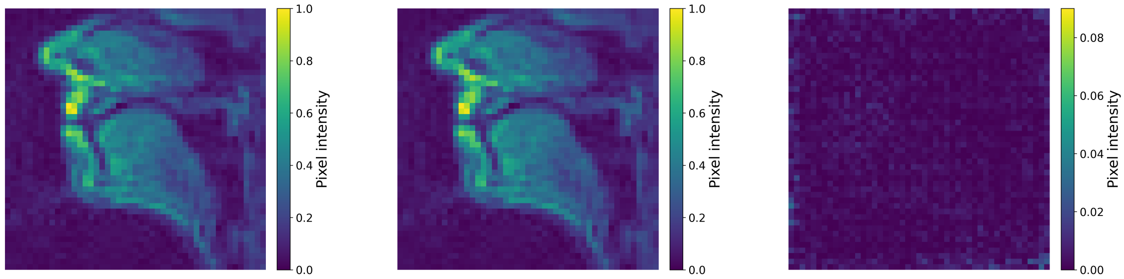


Figure 10: Average PLD loss on the test set: speaker-specific results are shown in blue, and speaker-invariant results are shown in orange.



(a) Original frame (b) Reconstructed frame (c) Difference

Figure 11: Visualization of reconstruction performance of the speaker-specific model on a single frame from participant F1. The MSE of the pixel values between the original and reconstructed frame was 3.2×10^{-4}.



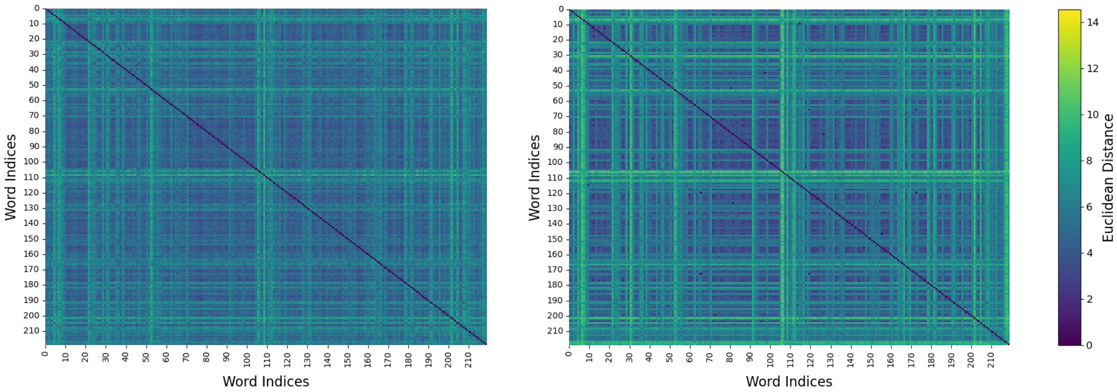
(a) Original frame (b) Reconstructed frame (c) Difference

Figure 12: Visualization of reconstruction performance of the speaker-invariant model on a single frame from participant F1. The MSE of the pixel values between the original and reconstructed frame was 1.9×10^{-5}.

4.2 Bottleneck Representations

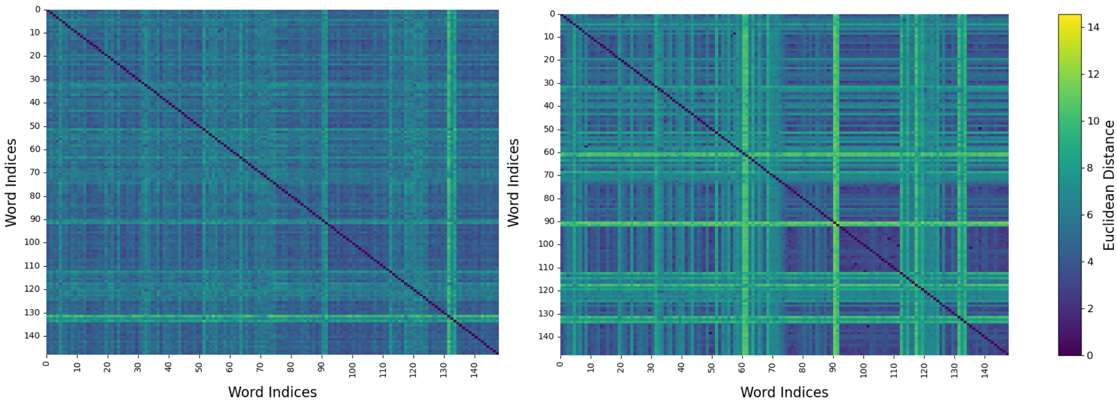
To gain better insight into the bottleneck representations from the autoencoder models, we generated 100-dimensional vectors representing word articulations for all rtMRI videos from the test set. Although we generated these vectors for all participants, in this section, we highlight the results of participants F1 and M5, demonstrating the impact of data availability, with participant M5 having less data. We highlight these results because they are representative of the similar overall results across all participants. In addition, we compare all vectors by correlating the speaker-specific and speaker-invariant distance vectors per participant, as described in Section 3.4.5. The similarity matrices for other participants are included in Appendix C.

Figures 13 and 14 compare the similarity matrices of participants F1 and M5, respectively, showing the speaker-specific model on the left and the speaker-invariant model on the right. The x and y axes represent the indices of the generated vectors. From these similarity matrices, it is evident that for both participants, the vectors show a similar structure. What stands out in the speaker-invariant similarity matrices is a wider range of distance values, with vectors either showing larger distances (yellow) or smaller distances (dark blue) than the speaker-specific similarity matrices. Since the test set per participant does not include the same data points, we cannot compare the similarity matrices between participants directly. Therefore, the similarity matrices can only be compared between speaker-specific and speaker-invariant models for the same participant, but not between different participants.



(a) Speaker-Specific F1 (b) Speaker-Invariant F1

Figure 13: Similarity matrices of bottleneck representations from participant F1: Each data point in the test set is compressed to a 100-dimensional vector. Distances between all vectors are calculated using Euclidean distance.



(a) Speaker-Specific M5 (b) Speaker-Invariant M5

Figure 14: Similarity matrices of bottleneck representations from participant M5: Each data point in the test set is compressed to a 100-dimensional vector. Distances between all vectors are calculated using Euclidean distance.

The correlation between the speaker-specific and speaker-invariant Euclidean distance vectors was tested for each participant using the Spearman rank-order correlation coefficient, as the data was not normally distributed. Positive correlations were found for each participant. Figure 15 presents the correlation coefficients for each participant, ranging from 0.63 (participant F4) to 0.93 (participant M4). The corresponding p-values for each correlation were significant ($p < 0.001$), with p-values very close to 0.0 due to the large number of data points in the Euclidean distance vectors (over 10,000 points). Interestingly, these correlations are related to the number of data points available. The lowest correlation was found for the Euclidean distance vectors of participant F4, who had the fewest videos available. Conversely, the highest correlation was found for participant M5, who had the most videos available, with 1,358 rtMRI videos for participant F4 and 2,518 rtMRI videos for participant M5 (See Figure 3.3 in Section 3.2.3).

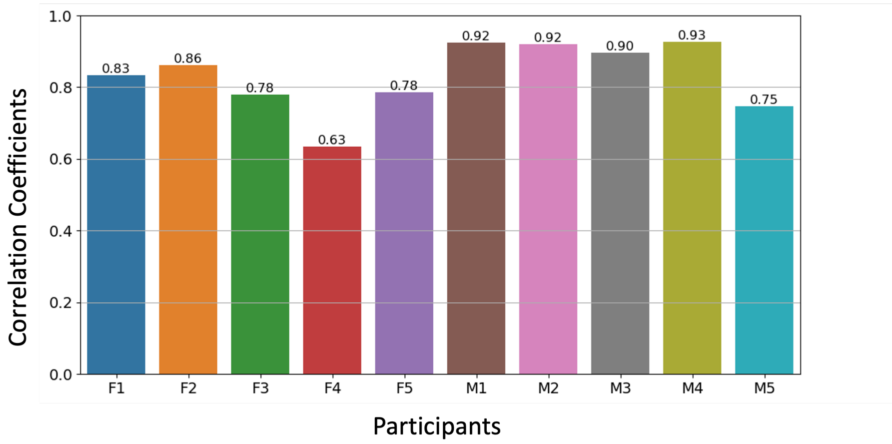
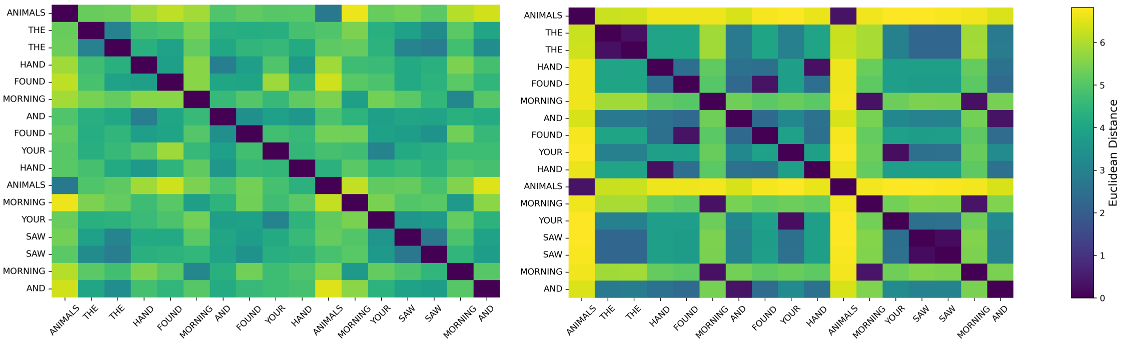


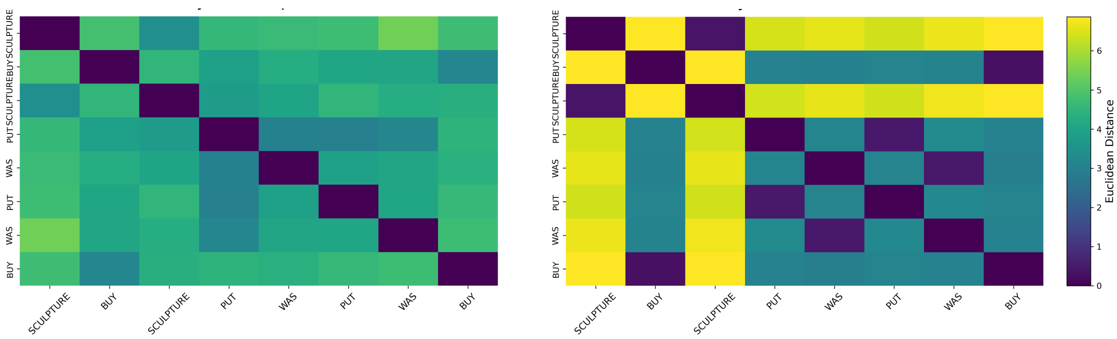
Figure 15: Correlations between speaker-specific and speaker-invariant similarity matrices per participant. The correlation coefficients are presented above each bar.

Furthermore, Figures 16 and 17 show specific bottleneck vectors for words that have multiple instances, meaning the same words were pronounced in different rtMRI videos. First, the same trend as before can be observed, namely that the bottleneck vectors show smaller and larger distances in the speaker-invariant similarity matrices (see Figures 16b and 17b). Specifically, the bottleneck vectors representing the same words have a small Euclidean distance that is close to zero (dark blue). A closer inspection of the words present in these figures shows that longer words, such as ‘animals’ (from participant F1) and ‘sculpture’ (from participant M5) have a large Euclidean distance compared to the other words, which is not observed in the speaker-specific similarity matrices for both participants. Another data point, ‘morning’ (see Figures 16a and 16b), can also be considered a long word. Although this word does not show a large difference in Euclidean distance between the two model categories, it does illustrate a smaller Euclidean distance for the same word instances. Additionally, what stands out in Figures 17a and 17b is that short words (3 characters) have a smaller Euclidean distance in the speaker-invariant model compared to the speaker-specific model. Figures 18 and 19 show the similarity matrices for bottleneck vectors of short words consisting of three characters. Again, these similarity matrices show that short words have a smaller Euclidean distance when trained with the speaker-invariant model. What is interesting about this comparison is that there is more variance in Euclidean distance between vectors in the speaker-specific model (see Figures 18a and 19a). These results suggest that the bottleneck vectors are less generalized based on word length in the speaker-specific model.



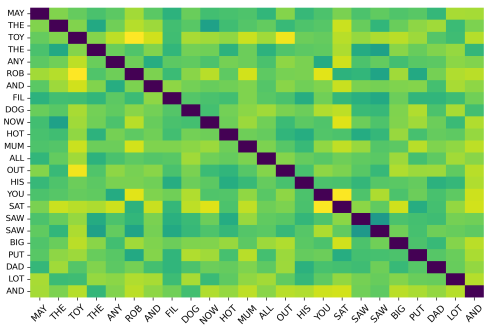
(a) Speaker-Specific F1 (b) Speaker-Invariant F1

Figure 16: Similarity matrices of bottleneck representations (zoomed-in) from participant F1: Euclidean distance between bottleneck vectors representing the same word label.

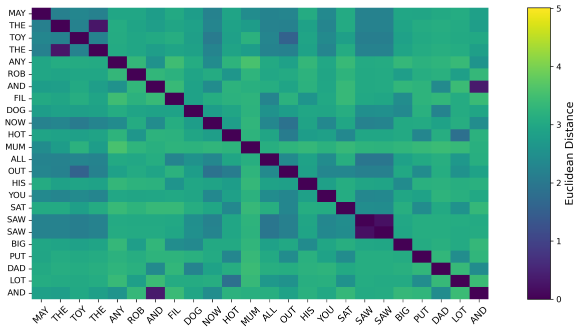


(a) Speaker-Specific M5 (b) Speaker-Invariant M5

Figure 17: Similarity matrices of bottleneck representations (zoomed-in) from participant M5: Euclidean distance between bottleneck vectors representing the same word label.

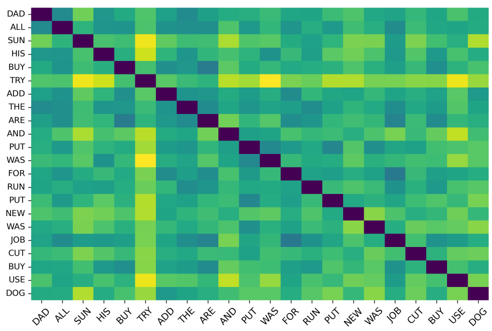


(a) Speaker-Specific F1

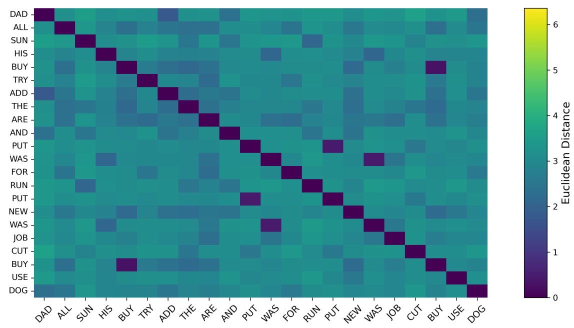


(b) Speaker-Invariant F1

Figure 18: Similarity matrices of bottleneck representations (zoomed-in) from participant F1: Euclidean distance between bottleneck vectors representing short words.



(a) Speaker-Specific M5



(b) Speaker-Invariant M5

Figure 19: Similarity matrices of bottleneck representations (zoomed-in) from participant M5: Euclidean distance between bottleneck vectors representing short words.

5 Discussion

In speech research, rtMRI has proven to be a promising and effective method for recording speech production data. This technique provides dynamic information of articulation movements during running speech production. However, data obtained from rtMRI can be complex and high-dimensional, making it challenging to analyse. This complexity is known as the 'curse of dimensionality', because when the number of dimensions (or features) increases, the difficulty in analyzing and interpreting the data also increases. To address this issue, we employed autoencoders to produce more compact feature vectors representing individual word articulations.

Another challenge in speech research is speaker specificity. Individuals differ in vocal tract morphology, with variations in shape and size of the lips, tongue, jaw, nasal cavity, and vocal cords. These anatomical differences affect how words are articulated. Due to this speaker variability, previous studies employing deep learning for rtMRI data have trained neural networks using a speaker-specific approach, meaning a separate model for each speaker (Csapó, 2020) (Yu *et al.*, 2021) (Stolwijk, 2022). However, a speaker-invariant approach supports speaker generalization by learning representations that capture features consistent across different speakers. Building on the work by Stolwijk (2022), this project aimed to investigate speaker generalization by comparing speaker-specific and speaker-invariant autoencoder models for reconstructing word articulations using rtMRI videos of the vocal tract.

5.1 Model Performance

The ConvGRU autoencoder architecture was able to effectively compress high-dimensional rtMRI data. After preprocessing, the data was reduced from 68 x 68 pixels per frame to 47 x 47 pixels per frame. The third dimension, representing the number of frames in a video, ranged from 5 to 35, resulting in at least 11,045 pixels (treating each pixel as a feature). This data was then compressed into a bottleneck vector of 1 x 100. The average MSE loss on the test set demonstrated that the autoencoder architecture effectively reconstructed the data for speaker-specific and speaker-invariant models for all ten participants. This finding highlights the compatibility of convolutional layers for handling image data and recurrent layers for processing sequential data.

Another important finding was that the speaker-invariant model improved the reconstruction performance and phoneme encoding for all ten participants. From the total loss values reported in Appendix B, it is evident that the speaker-invariant model reduces the total loss by a factor of approximately 10 compared to the speaker-specific models. The total loss values include both MSE and PLD loss values. These significant improvements in performance can be attributed to two main factors. First, the speaker-invariant model is trained on a much larger dataset, increasing from approximately 2,000 data points to 20,000 data points. Second, the inclusion of data from different speakers introduces more variation, allowing the model to generalize better across different individuals. Although the speaker-invariant model was tested on data specific to individual participants, the increased data and variation improved its generalizability to unseen data.

5.2 Reconstruction Performance from Literature

The previous study conducted by Stolwijk (2022) for their master's thesis, discussed in Section 2.5, employed the ConvGRU autoencoder model with a speaker-specific approach using rtMRI data from participant F1. In this study, one experiment focused on cross-participant transferability, meaning the model was trained on data from a specific participant (F1) and tested on data from other participants (F1 to F5 and M1 to M5). Initial testing showed poor performance on the reconstruction loss, with loss values five times higher compared to participant F1. To improve the reconstruction performance, the speaker-specific model, initially trained on data from F1, was fine-tuned by adding data from the specific participant being tested. This fine-tuning improved the model's reconstruction loss to be almost similar to that of participant F1 (loss: 10), with loss values ranging from 8.78 (M1) to 22.23 (M5). Since the pixel values of the rtMRI videos in these experiments were not normalized, direct comparison with the loss values is not possible. Additionally, the test set of these experiments may have included other rtMRI videos, and thus, other word labels.

However, we can make relative comparisons between the previous and current studies. The previous study showed that participant M5 had the highest reconstruction loss when testing the speaker-specific model trained on data from F1. After fine-tuning by adding data from participant M5, the loss decreased but was still the highest compared to other participants. The same procedure was applied to other participants, where the model was initially trained with data from participant F1 and then fine-tuned by adding data from the specific participant being tested.

In the current study, we trained speaker-specific models for all participants. Consistent with the previous study (see Appendix B), the model performance of participant M5 showed the highest reconstruction loss compared to the other participants. For the speaker-invariant model, which combines data from all participants, it was still observed that the reconstruction loss for participant M5 was the highest among participants. However, the loss was substantially lower, decreasing from 3.7×10^{-4} for the speaker-specific model to 5.3×10^{-5} for the speaker-invariant model. As discussed by Stolwijk (2022), this could be due to noise in the data of participant M5. Furthermore, there was less data available for participant M5 compared to other participants, except for participant F4 (see Table 3.3). Since participant F4 also had less data available, we observed that the reconstruction loss of participant F4 was the second highest for the speaker-specific model, but this was not the case for the speaker-invariant model.

5.3 Bottleneck Representations

The bottleneck representations are a compressed form of word articulations in the rtMRI videos. These 100-dimensional vectors provide insights into how different models process and encode speech data. We computed the Euclidean distance between each bottleneck vector in the test set and plotted the results as similarity matrices. The most notable finding from the comparison between the speaker-specific and speaker-invariant models was that the similarity matrices showed similar structures, with the speaker-invariant model having both smaller and larger Euclidean distance values. The correlation between the speaker-specific and speaker-invariant models for each participant revealed significant positive correlations, indicating a high degree of similarity in the processing and representation of word articulations. These strong correlations suggest that both models capture the essential features of the speech data.

A possible explanation for the difference in model performance is the difference in Euclidean distances observed in the similarity matrices. Bottleneck vectors representing the same words but from different data points (repeated words) showed low to zero Euclidean distances in the speaker-invariant model. Data points representing the repeated words and short words showed greater similarity in the speaker-invariant model compared to the speaker-specific model. Since the speaker-invariant model is trained on almost ten times more data, it can better learn to represent similar words consistently. We focused on short words of three characters because the similarity matrices of the repeated words indicated that short words had smaller Euclidean distances than longer words in the speaker-invariant model.

Another finding from the correlations between the similarity matrices of speaker-specific and speaker-invariant models is the impact of data quantity. The strength of correlations varied with the data quantity. Specifically, the correlations were weaker when less data were available, as observed for participants F4 and M5 in Figure 15. The bottleneck representations also include the phonemes of the words spoken in the rtMRI videos. Participants F4 and M5 had the highest PLD loss among all participants for the speaker-specific model (see Figure 10). When training with much more data, the PLD loss significantly decreased to the second lowest and lowest loss values, respectively, for F4 and M5 (shown in Appendix B). This highlights the importance of data quantity.

5.4 Limitations

5.4.1 Data

The USC-TIMIT database originally consisted of rtMRI videos of sentences, which were then segmented into individual words. A limitation of this method is that the words in these sentences are dependent on each other, introducing coarticulation effects. Consequently, the pronunciation of a word is affected by the words spoken before and after it. Another detail of the segmentation method is that the transcription files, which consist of the start and end times of when specific words were spoken, are not always precise. These times were recorded with only two decimal places and may be affected by the frame rate of 23.18 frames per second. This limited temporal resolution can cause overlap, especially with short words, leading to parts of a word's articulation being incorrectly assigned to the wrong video segment. Higher frame rates and more precise transcription could improve the word segmentation. Another solution would be to record individual words to prevent coarticulation.

Another limitation is the data quality. Specifically, the videos of participant M5 showed noise in the frames. Manually checking the videos would be very time-consuming, as there are approximately 20,000 videos in total. An interesting and possible solution would be to apply denoising autoencoders to improve the data quality. In addition to the noise in the video frames, there were also transcription errors, as mentioned by Stolwijk (2022). These errors were difficult to manually check both because of the size of the dataset and due to the low audio quality. This issue arose because the audio was acquired in a MRI scanner.

Two participants had less data available due to missing frames in the videos. This highlights the impact of data quantity. For future research, it might be possible to apply data augmentation. This involves creating new training data by transforming the existing data. Specific transformation operations include shifting the video frames slightly in different directions (left, right, up, or down) and zooming in or out. These transformations help the model by exposing it to different perspectives and scales of the same data, thereby improving its ability to generalize.

5.4.2 Model Training

Since this study builds on the foundation of a previous master's thesis project by Stolwijk (2022), we adopted a similar experimental setup, given that the same autoencoder architecture was used. Consequently, similar hyperparameters were employed for model training, including weight decay, batch size, and the scaling weight for PLD loss.

However, because the current study employed a speaker-invariant approach and normalized pixel values in the video data, we adjusted certain hyperparameters such as the number of epochs and the learning rate. We also implemented a learning rate scheduler, with the initial learning rate determined through hyperparameter optimization on the validation loss. Additionally, we employed early stopping to halt training when the validation loss did not decrease, ensuring efficient training and preventing overfitting.

In addition to these adjustments, hyperparameters such as the PLD weight and batch size for mini-batch training could still be optimized for the current experimental setup. The PLD weight is particularly important because it influences how much phoneme information the model encodes. Optimizing this weight could improve the model's ability to capture phonetic details. Similarly, adjusting the batch size could enhance model performance, especially since the speaker-invariant model trains on a much larger dataset than the speaker-specific models. Due to time constraints, it was not possible to explore these optimizations in the current study. However, future research should consider optimizing these hyperparameters to further enhance model performance.

5.5 Future Work

In future investigations, the speaker-invariant model could potentially be extended to other languages. Currently, the experimental setup uses phonemes from American English, so adjustments would be necessary for other languages. This master's thesis project was conducted in collaboration with the Utrecht-BCI Lab, making it particularly beneficial to use a Dutch dataset, given that the BCI research is primarily aimed at Dutch speakers. The speaker-invariant model trained on American English words could serve as a pre-trained model for a Dutch dataset, as both languages share similarities in phonemes. Specifically, both Dutch and (American) English include a set of common phonemes, which means the model's learned features for these phonemes can be beneficial for processing Dutch phonemes. However, Dutch has unique phonemes that the model might not fully capture initially. Therefore, while the pre-trained model offers a strong foundation and can reduce the time required for training, some fine-tuning with Dutch data may be necessary to adjust for these language-specific differences.

An important detail is that although Dutch rtMRI data was available from the Utrecht-BCI Lab, it had not yet been preprocessed as thoroughly as the USC-TIMIT dataset. Due to time constraints, we were unable to include the Dutch dataset in this project.

Another way to include more variability in word articulations is by including speech data that expresses different emotions, as speech movements are dependent on the emotion conveyed. The study by Pandey and Arif (2021) found that different regions of the vocal tract are affected by various emotions, such as neutral, happy, angry, and sad. By adding more variation in the data, the model is exposed to a wider range of articulation patterns, which can improve feature representation and model generalization.

Lastly, additional research is needed to better understand the relationship between articulation patterns and neural representations. The obtained bottleneck vectors of word articulations could be compared with the corresponding neural representations, specifically neural activity from the sensorimotor cortex representing movements of the vocal tract during speech production (Chartier *et al.*, 2018).

6 Conclusion

This project was undertaken to design a speaker-invariant model to investigate speech production of individual words. The high-dimensional rtMRI video data were compressed using a convolutional and recurrent autoencoder architecture. The model was evaluated based on its reconstruction performance and phoneme encoding, compared to a speaker-specific model. To further analyze the obtained bottleneck vectors generated by the autoencoder, the Euclidean distance between these vectors was calculated, resulting in more interpretable similarity matrices. This study has identified that the speaker-invariant model leverages two key aspects: higher data quantity and increased data variability. These aspects result in lower reconstruction loss and more accurate phoneme encoding, as demonstrated by the significantly reduced PLD loss.

A limitation of this study is that the data were originally recorded as sentences rather than individual words. This required preprocessing steps to split the videos into individual words, which possibly introduced errors into the data. The model's performance could be improved by using more reliable transcription methods to ensure that the video frames accurately correspond to the correct word labels, without interference from frames including other words.

The findings from this study are relevant to the development of speech-BCIs that focus on word decoding from brain activity of attempted speech in combination with deep learning. A natural progression of this work is to analyze neural representations of articulatory movements and compare these representations to word articulations. Neural networks have been increasingly utilized in recent studies to analyze brain data and advance the development of BCI for more natural and efficient communication. Therefore, as data dimensionality increases in this field, autoencoder architectures that rely on convolutional and recurrent layers can be effectively applied to reduce dimensionality in both image and sequential data.

Acronyms

3D-CNN Three-dimensional Convolutional Neural Network. 6, 7

ALS Amyotrophic Lateral Sclerosis. ii, 1, 4

BCI brain-computer interface. 1–4, 28, 29
BCIs brain-computer interfaces. ii, 1, 2, 29

CMUDict CMU Pronouncing Dictionary. 10
CNN convolutional neural network. 5, 6
CNNs Convolutional neural networks. 12
ConvGRU Convolutional Gated Recurrent Unit. vi, 6, 7, 12–14, 25

ECoG electrocorticography. 1
EEG electroencephalography. 1
EMA electromagnetic articulography. 2, 4, 5, 8

GRU Gated Recurrent Unit. 13, 14

LIS locked-in syndrome. ii, 1, 4
LSTM long short-term memory. 5

MRI Magnetic Resonance Imaging. 4
MSE mean squared error. vi, viii, 7, 15–20, 25, 40

NLTK Natural Language Tool-Kit. 10

PD Parkinson's Disease. 1
PLD Phonemic Levenshtein Distance. vi, viii, 7, 15, 16, 18, 19, 25–27, 29, 40

ReLU rectified linear unit. 13, 14
RNN recurrent neural network. 5, 6, 13
RNNs recurrent neural networks. 13, 15
rtMRI real-time Magnetic Resonance Imaging. ii, vi–viii, 2–6, 8–10, 12, 15, 18, 20, 22, 25–29, 36–39

Bibliography

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *5*, 157–66. https://doi.org/10.1109/72.279181

Berger, A. (2002). How does it work?: Magnetic resonance imaging. *BMJ: British Medical Journal*, *324*(7328), 35.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: Analyzing text with the natural language toolkit.* " O'Reilly Media, Inc."

Bruno, M.-A., Schnakers, C., Damas, F., Pellas, F., Lutte, I., Bernheim, J., Majerus, S., Moonen, G., Goldman, S., & Laureys, S. (2009). Locked-in syndrome in children: Report of five cases and review of the literature. *Pediatric neurology*, *41*(4), 237–246.

Carnegie Mellon University. (1998). The carnegie mellon university pronouncing dictionary (version cmudict.0.6). http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Ceslis, A., Argall, R., Henderson, R. D., McCombe, P. A., & Robinson, G. A. (2020). The spectrum of language impairments in amyotrophic lateral sclerosis. *Cortex*, *132*, 349–360.

Chartier, J., Anumanchipalli, G. K., Johnson, K., & Chang, E. F. (2018). Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. *Neuron*, *98*(5), 1042–1054.

Cho, K., Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. https://doi.org/10.3115/v1/W14-4012

Chong, Y. S., & Tay, Y. H. (2017). Abnormal event detection in videos using spatiotemporal autoencoder. *Advances in Neural Networks-ISNN 2017: 14th International Symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21–26, 2017, Proceedings, Part II 14*, 189–196.

Conant, D. F., Bouchard, K. E., Leonard, M. K., & Chang, E. F. (2018). Human sensorimotor cortex control of directly measured vocal tract movements during vowel production. *Journal of Neuroscience*, *38*(12), 2955–2966.

Csapó, T. G. (2020). Speaker dependent articulatory-to-acoustic mapping using real-time mri of the vocal tract. *arXiv preprint arXiv:2008.00889*.

Dawson, K. M., Tiede, M. K., & Whalen, D. (2016). Methods for quantifying tongue shape and complexity using ultrasound imaging. *Clinical linguistics & phonetics*, *30*(3-5), 328–344.

Felgoise, S. H., Zaccheo, V., Duff, J., & Simmons, Z. (2016). Verbal communication impacts quality of life in patients with amyotrophic lateral sclerosis. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, *17*(3-4), 179–183.

Francis, D., Sherman, A., Hovis, K., Bonnet, K., Schlundt, D., Garrett, C., & Davies, L. (2018). Life experience of patients with unilateral vocal fold paralysis. *JAMA otolaryngology–head & neck surgery*, *144*. https://doi.org/10.1001/jamaoto.2018.0067

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Halan, T., Ortiz, J. F., Reddy, D., Altamimi, A., Ajibowo, A. O., & Fabara, S. P. (2021). Locked-in syndrome: A systematic review of long-term management and prognosis. *Cureus*, *13*(7).

He, B., Yuan, H., Meng, J., & Gao, S. (2020). Brain–computer interfaces. *Neural engineering*, 131–183.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231. https://doi.org/10.1109/TPAMI.2012.59

Jurafsky, D., & Martin, J. H. (2024). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (3rd ed. draft). https://web.stanford.edu/~jurafsky/slp3/

Katz, W. F., Bharadwaj, S. V., & Carstens, B. (1999). Electromagnetic articulography treatment for an adult with broca's aphasia and apraxia of speech. *Journal of Speech, Language, and Hearing Research*, *42*(6), 1355–1366.

Kim, J., Lammert, A. C., Kumar Ghosh, P., & Narayanan, S. S. (2014). Co-registration of speech production datasets from electromagnetic articulography and real-time magnetic resonance imaging. *The Journal of the Acoustical Society of America*, *135*(2), EL115–EL121.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kose, O., & Saraclar, M. (2021). Multimodal representations for synchronized speech and real-time mri video processing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *PP*, 1–1. https://doi.org/10.1109/TASLP.2021.3084099

Krizhevsky, A., Hinton, G., *et al.* (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Levenshtein, V. I., *et al.* (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, *10*(8), 707–710.

Lulé, D., Zickler, C., Häcker, S., Bruno, M.-A., Demertzi, A., Pellas, F., Laureys, S., & Kübler, A. (2009). Life can be worth living in locked-in syndrome. *Progress in brain research*, *177*, 339–351.

Masrori, P., & Van Damme, P. (2020). Amyotrophic lateral sclerosis: A clinical review. *European journal of neurology*, *27*(10), 1918–1929.

Menghani, G. (2023). Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, *55*(12), 1–37.

Metzger, S. L., Littlejohn, K. T., Silva, A. B., Moses, D. A., Seaton, M. P., Wang, R., Dougherty, M. E., Liu, J. R., Wu, P., Berger, M. A., *et al.* (2023). A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, *620*(7976), 1037–1046.

Mohammadi, S. H., & Kain, A. (2014). Voice conversion using deep neural networks with speaker-independent pre-training. *2014 IEEE Spoken Language Technology Workshop (SLT)*, 19–23. https://doi.org/10.1109/SLT.2014.7078543

Moses, D. A., Metzger, S. L., Liu, J. R., Anumanchipalli, G. K., Makin, J. G., Sun, P. F., Chartier, J., Dougherty, M. E., Liu, P. M., Abrams, G. M., *et al.* (2021). Neuroprosthesis for decoding speech in a paralyzed person with anarthria. *New England Journal of Medicine*, *385*(3), 217–227.

Narayanan, S., Nayak, K., Lee, S., Sethy, A., & Byrd, D. (2004). An approach to real-time magnetic resonance imaging for speech production. *The Journal of the Acoustical Society of America*, *115*(4), 1771–1776.

Narayanan, S., Toutios, A., Ramanarayanan, V., Lammert, A., Kim, J., Lee, S., Nayak, K., Kim, Y.-C., Zhu, Y., Goldstein, L., *et al.* (2014). Real-time magnetic resonance imaging and electromagnetic articulography database for speech production research (tc). *The Journal of the Acoustical Society of America*, *136*(3), 1307–1311.

Pandey, L., & Arif, A. S. (2021). Silent speech and emotion recognition from vocal tract shape dynamics in real-time mri. https://arxiv.org/abs/2106.08706

Parrot, M., Millet, J., & Dunbar, E. (2020). Independent and automatic evaluation of speaker-independent acoustic-to-articulatory reconstruction. *Interspeech 2020-21st Annual Conference of the International Speech Communication Association*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch.

Rabbani, Q., Milsap, G., & Crone, N. E. (2019). The potential for a speech brain–computer interface using chronic electrocorticography. *Neurotherapeutics*, *16*(1), 144–165.

Rebernik, T., Jacobi, J., Jonkers, R., Noiray, A., & Wieling, M. (2021). A review of data collection practices using electromagnetic articulography. *Laboratory Phonology*, *12*(1), 6.

Rezeika, A., Benda, M., Stawicki, P., Gembler, F., Saboor, A., & Volosyak, I. (2018). Brain–computer interface spellers: A review. *Brain sciences*, *8*(4), 57.

Rousseau, M.-C., Baumstarck, K., Alessandrini, M., Blandin, V., Billette de Villemeur, T., & Auquier, P. (2015). Quality of life in patients with locked-in syndrome: Evolution over a 6-year period. *Orphanet journal of rare diseases*, *10*, 1–8.

Ruoppolo, G., Schettino, I., Frasca, V., Giacomelli, E., Prosperini, L., Cambieri, C., Roma, R., Greco, A., Mancini, P., De Vincentiis, M., *et al.* (2013). Dysphagia in amyotrophic lateral sclerosis: Prevalence and clinical findings. *Acta Neurologica Scandinavica*, *128*(6), 397–401.

Ruthven, M., Miquel, M. E., & King, A. P. (2021). Deep-learning-based segmentation of the vocal tract and articulators in real-time magnetic resonance images of speech. *Computer Methods and Programs in Biomedicine*, *198*, 105814.

Schalk, G., & Leuthardt, E. C. (2011). Brain-computer interfaces using electrocorticographic signals. *IEEE reviews in biomedical engineering*, *4*, 140–154.

Sellers, E. W., Ryan, D. B., & Hauser, C. K. (2014). Noninvasive brain-computer interface enables communication after brainstem stroke. *Science translational medicine*, *6*(257), 257re7–257re7.

Simonyan, K., & Horwitz, B. (2011). Laryngeal motor cortex and control of speech in humans. *The Neuroscientist*, *17*(2), 197–208.

Smith, E., & Delargy, M. (2005). Locked-in syndrome. *Bmj*, *330*(7488), 406–409.

Smith, K. M., & Caplan, D. N. (2018). Communication impairment in parkinson's disease: Impact of motor and cognitive symptoms on speech and language. *Brain and language*, *185*, 38–46.

Stolwijk, E. (2022). *Towards speech-based brain-computer interfaces: Finding most distinguishable word articulations with autoencoders* [Published Master's thesis, Utrecht University]. UtrechtUniversityStudentThesesRepository.https://studenttheses.uu.nl/handle/20.500.12932/43475.

Toutios, A., Byrd, D., Goldstein, L., & Narayanan, S. (2019). Advances in vocal tract imaging and analysis. In *The routledge handbook of phonetics* (pp. 34–50). Routledge.

Toutios, A., & Narayanan, S. S. (2016). Advances in real-time magnetic resonance imaging of the vocal tract for speech science and technology research. *APSIPA Transactions on Signal and Information Processing*, *5*, e6.

Toutios, A., Sorensen, T., Somandepalli, K., Alexander, R., & Narayanan, S. S. (2016). Articulatory synthesis based on real-time magnetic resonance imaging data. *Interspeech*, 1492–1496.

Trail, M., Fox, C., Ramig, L. O., Sapir, S., Howard, J., & Lai, E. C. (2005). Speech treatment for parkinson's disease. *NeuroRehabilitation*, *20*(3), 205–221.

Van Leeuwen, K., Bos, P., Trebeschi, S., van Alphen, M. J., Voskuilen, L., Smeele, L. E., van der Heijden, F., van Son, R., *et al.* (2019). Cnn-based phoneme classifier from vocal tract mri learns embedding consistent with articulatory topology. *Interspeech*, 909–913.

van Leeuwen, K., Bos, P., Trebeschi, S., Alphen, M., Voskuilen, L., Smeele, L., Van der Heijden, F., & van Son, R. (2019). Cnn-based phoneme classifier from vocal tract mri learns embedding consistent with articulatory topology, 909–913. https://doi.org/10.21437/Interspeech.2019-1173

Vansteensel, M. J., Pels, E. G., Bleichner, M. G., Branco, M. P., Denison, T., Freudenburg, Z. V., Gosselaar, P., Leinders, S., Ottens, T. H., Van Den Boom, M. A., *et al.* (2016). Fully implanted brain–computer interface in a locked-in patient with als. *New England Journal of Medicine*, *375*(21), 2060–2066.

Värbu, K., Muhammad, N., & Muhammad, Y. (2022). Past, present, and future of eeg-based bci applications. *Sensors*, *22*(9), 3331.

Willett, F. R., Kunz, E. M., Fan, C., Avansino, D. T., Wilson, G. H., Choi, E. Y., Kamdar, F., Glasser, M. F., Hochberg, L. R., Druckmann, S., *et al.* (2023). A high-performance speech neuroprosthesis. *Nature*, *620*(7976), 1031–1036.

Wilson, I. (2014). Using ultrasound for teaching and researching articulation. *Acoustical Science and Technology*, *35*(6), 285–289.

Wrench, A. (2000). A multi-channel/multi-speaker articulatory database for continuous speech recognition research. *Univ. Saarland, Res. Rep.*, *5*, 1–13.

Yu, Y., Shandiz, A. H., & Tóth, L. (2021). Reconstructing speech from real-time articulatory mri using neural vocoders. *2021 29th European Signal Processing Conference (EUSIPCO)*, 945–949.

Appendix

A Frame distribution histograms

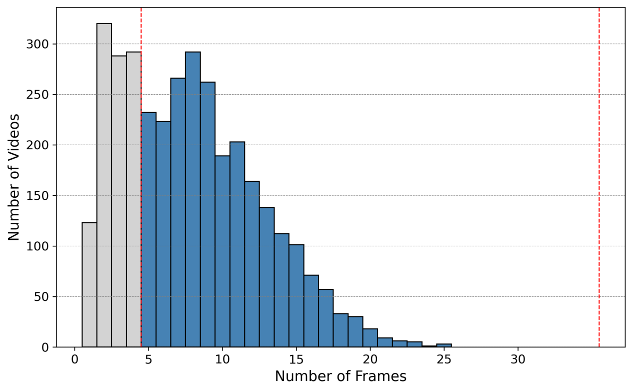


Figure 20: Frame distribution histogram of rtMRI data from participant F2.

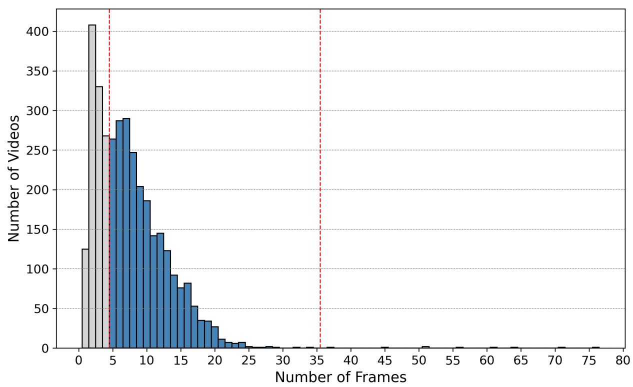


Figure 21: Frame distribution histogram of rtMRI data from participant F3.

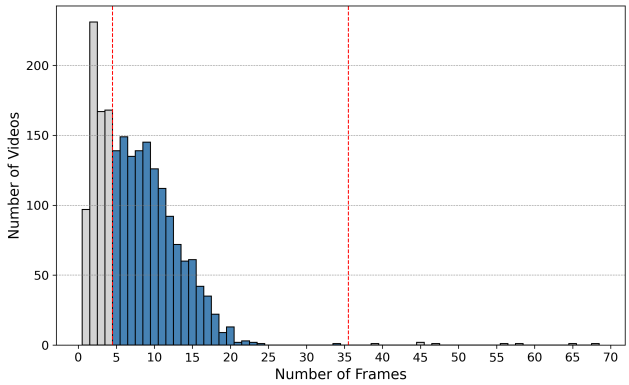


Figure 22: Frame distribution histogram of rtMRI data from participant F4.

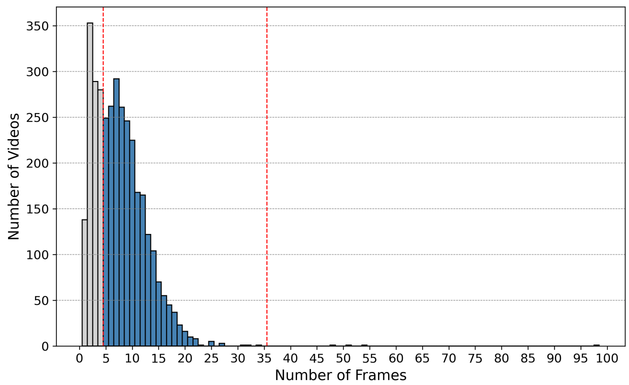


Figure 23: Frame distribution histogram of rtMRI data from participant F5.

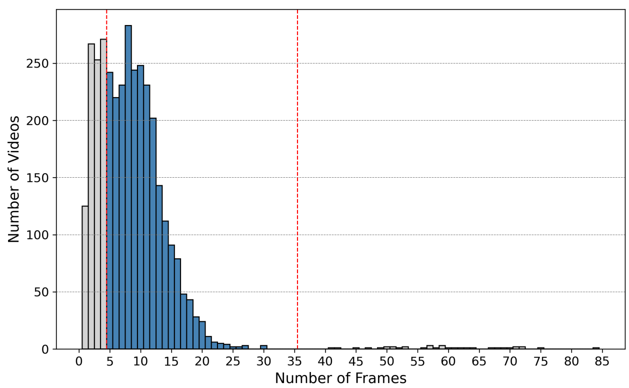


Figure 24: Frame distribution histogram of rtMRI data from participant M1.

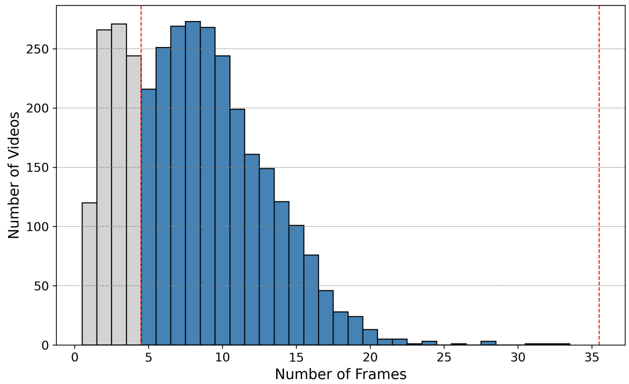


Figure 25: Frame distribution histogram of rtMRI data from participant M2.

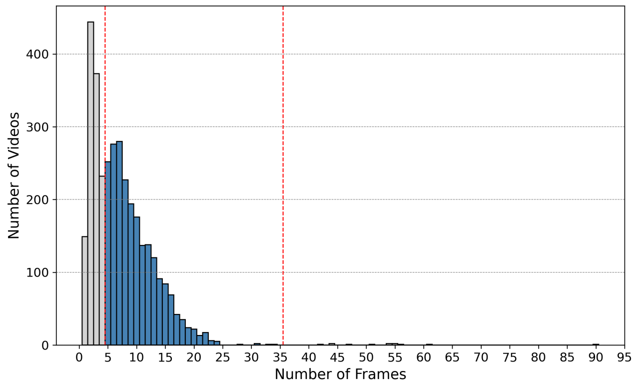


Figure 26: Frame distribution histogram of rtMRI data from participant M3.

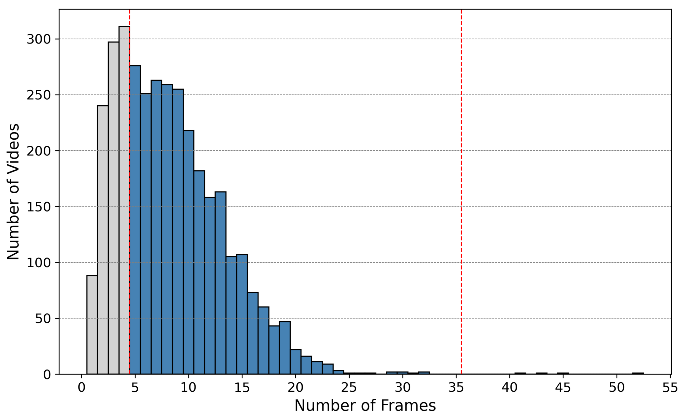


Figure 27: Frame distribution histogram of rtMRI data from participant M4.

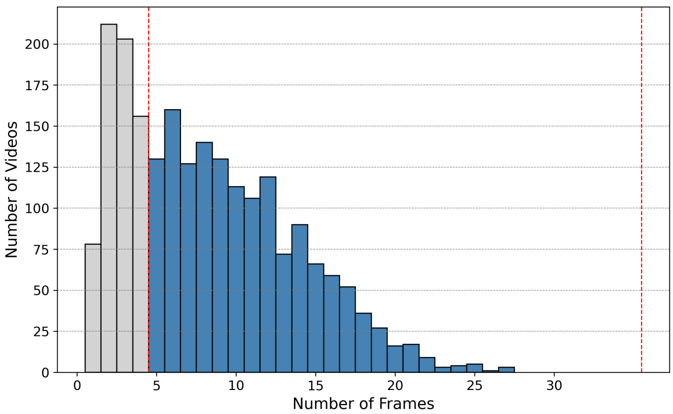


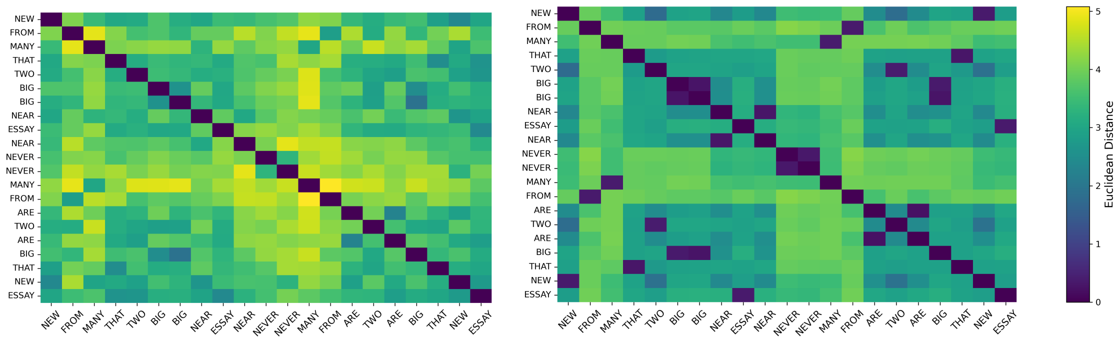
Figure 28: Frame distribution histogram of rtMRI data from participant M5.

B Reconstruction and Phoneme Loss

Participant	Speaker-specific				Speaker-invariant			
	Total	MSE	PLD	Epoch	Total	MSE	PLD	Epoch
F1	0.0068	0.00030	0.0065	93	0.000662	0.000021	0.00064	-
F2	0.0071	0.00017	0.0070	124	0.000690	0.000032	0.00066	-
F3	0.0074	0.00018	0.0072	86	0.000660	0.000027	0.00063	-
F4	0.0092	0.00033	0.0089	130	0.000650	0.000026	0.00063	-
F5	0.0083	0.00016	0.0081	114	0.000690	0.000024	0.00067	-
M1	0.0050	0.000066	0.0049	194	0.000810	0.000019	0.00079	-
M2	0.0053	0.00014	0.0051	197	0.000680	0.000029	0.00066	-
M3	0.0062	0.000080	0.0061	194	0.000950	0.000025	0.00092	-
M4	0.0043	0.00012	0.0042	191	0.000830	0.000024	0.00080	-
M5	0.0110	0.00037	0.011	134	0.000670	0.000053	0.00062	-
All	-	-	-	-	-	-	-	88

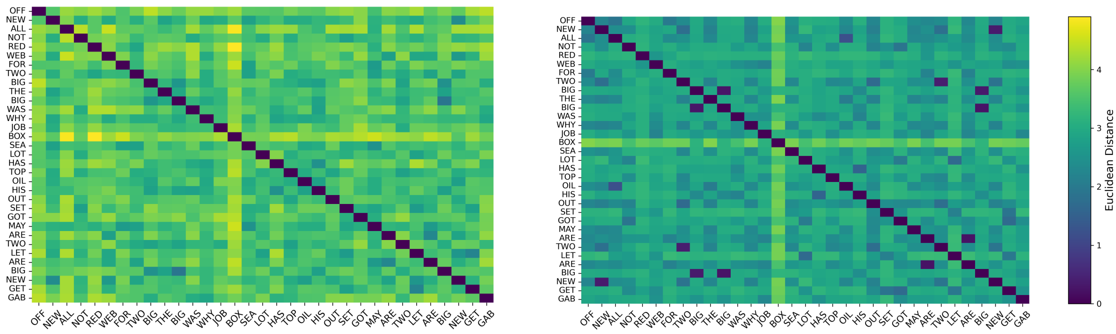
Table B.1: Speaker-Specific and Speaker-Invariant Results: Average MSE and PLD loss values tested specific to a participant. The epochs listed indicate when the validation loss reached its lowest point; this model was used to test the performance on unseen data.

C Bottleneck Representations



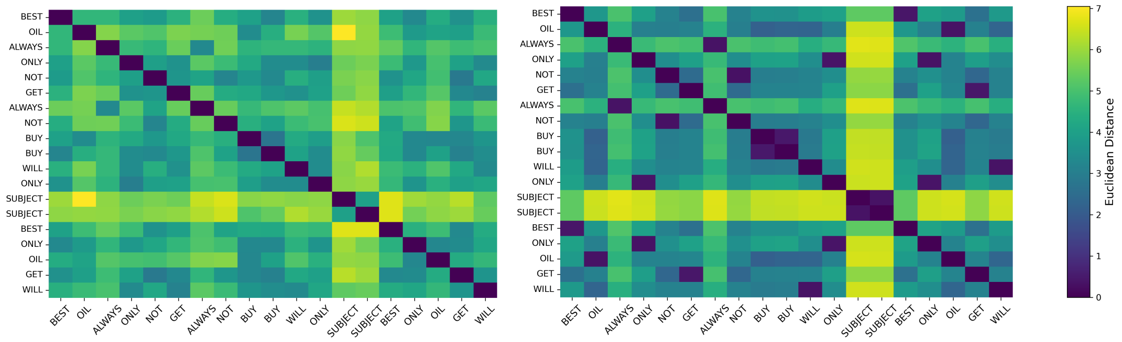
(a) Speaker-Specific F2 (b) Speaker-Invariant F2

Figure 29: Similarity matrices of bottleneck representations (zoomed-in) from participant F2: Euclidean distance between bottleneck vectors representing the same word label.



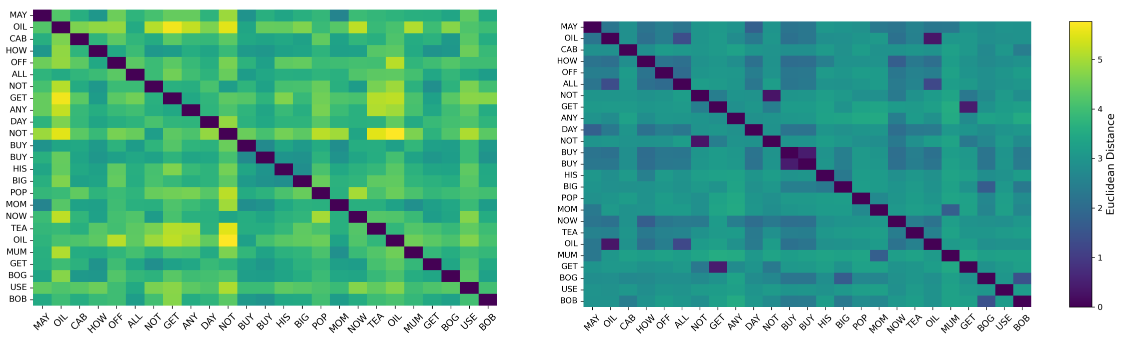
(a) Speaker-Specific F2 (b) Speaker-Invariant F2

Figure 30: Similarity matrices of bottleneck representations (zoomed-in) from participant F2: Euclidean distance between bottleneck vectors representing short words.



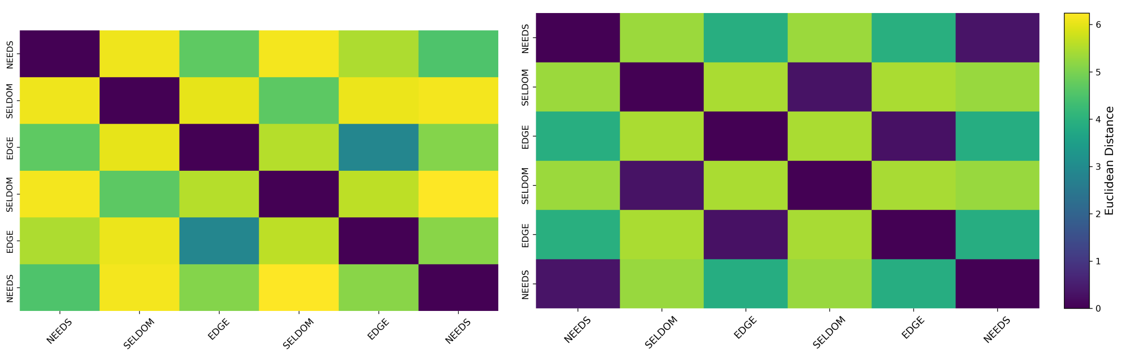
(a) Speaker-Specific F3 (b) Speaker-Invariant F3

Figure 31: Similarity matrices of bottleneck representations (zoomed-in) from participant F3: Euclidean distance between bottleneck vectors representing the same word label.



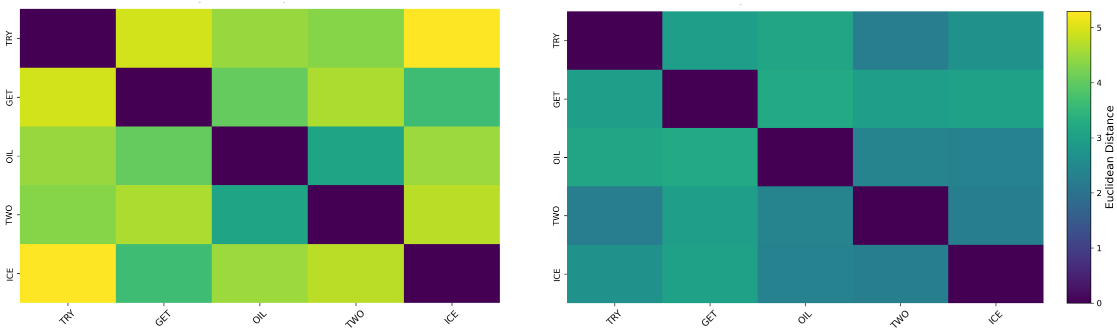
(a) Speaker-Specific F3 (b) Speaker-Invariant F3

Figure 32: Similarity matrices of bottleneck representations (zoomed-in) from participant F3: Euclidean distance between bottleneck vectors representing short words.



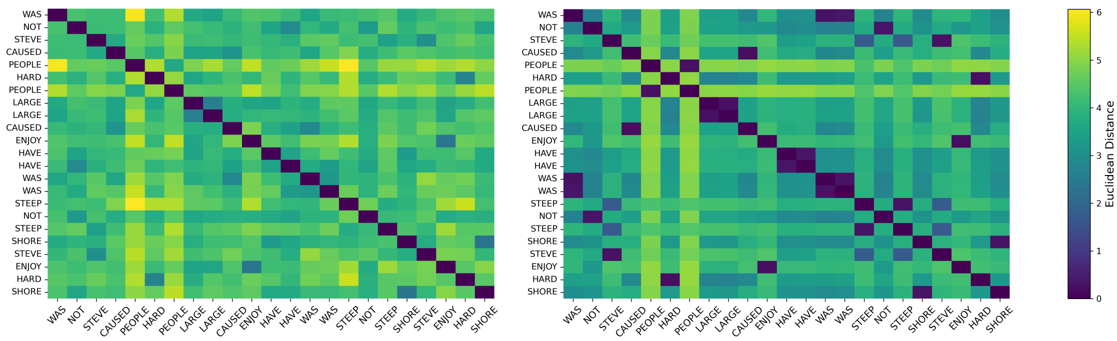
(a) Speaker-Specific F4 (b) Speaker-Invariant F4

Figure 33: Similarity matrices of bottleneck representations (zoomed-in) from participant F4: Euclidean distance between bottleneck vectors representing the same word label.



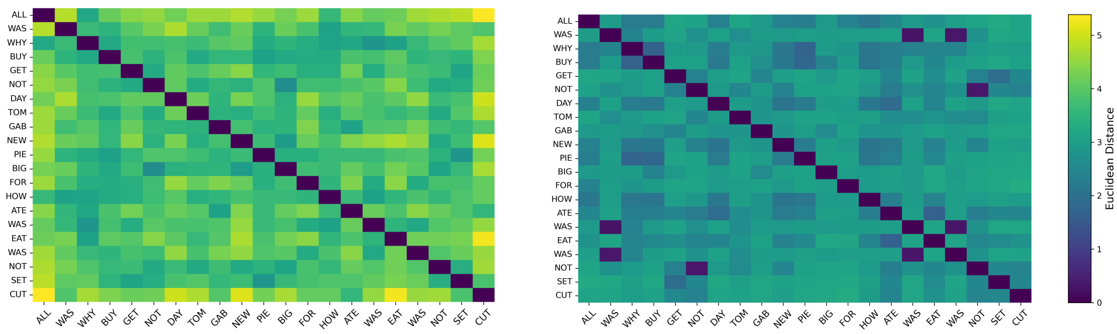
(a) Speaker-Specific F4 (b) Speaker-Invariant F4

Figure 34: Similarity matrices of bottleneck representations (zoomed-in) from participant F4: Euclidean distance between bottleneck vectors representing short words.



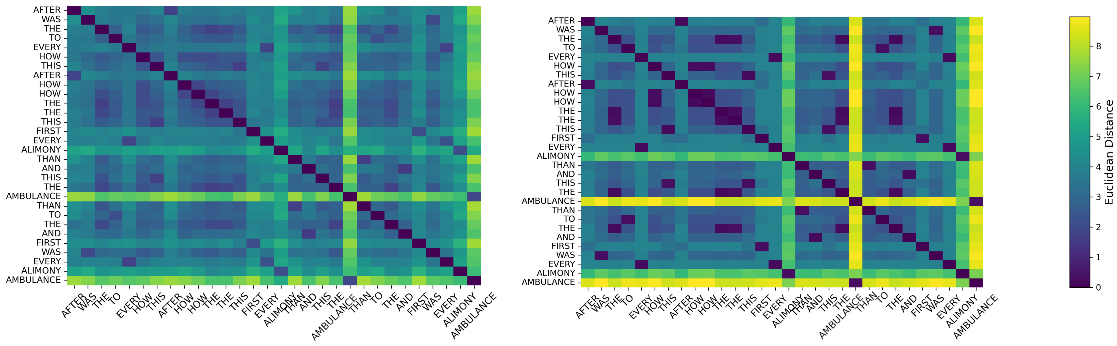
(a) Speaker-Specific F5 (b) Speaker-Invariant F5

Figure 35: Similarity matrices of bottleneck representations (zoomed-in) from participant F5: Euclidean distance between bottleneck vectors representing the same word label.



(a) Speaker-Specific F5 (b) Speaker-Invariant F5

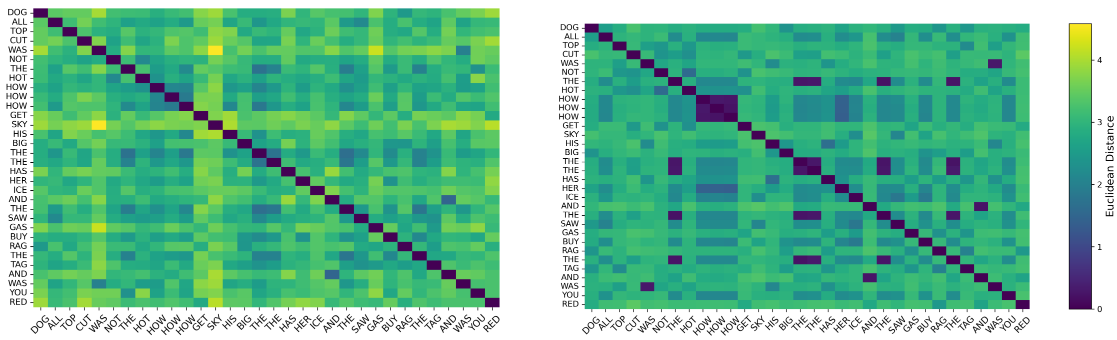
Figure 36: Similarity matrices of bottleneck representations (zoomed-in) from participant F5: Euclidean distance between bottleneck vectors representing short words.



(a) Speaker-Specific M1

(b) Speaker-Invariant M1

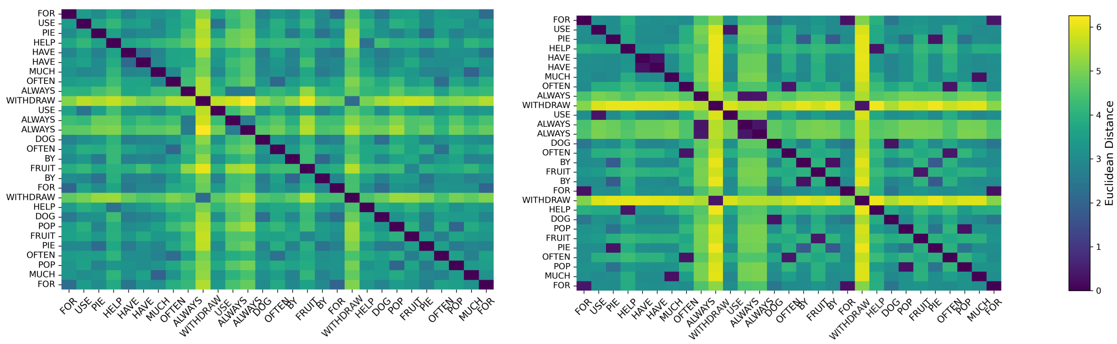
Figure 37: Similarity matrices of bottleneck representations (zoomed-in) from participant M1: Euclidean distance between bottleneck vectors representing the same word label.



(a) Speaker-Specific M1

(b) Speaker-Invariant M1

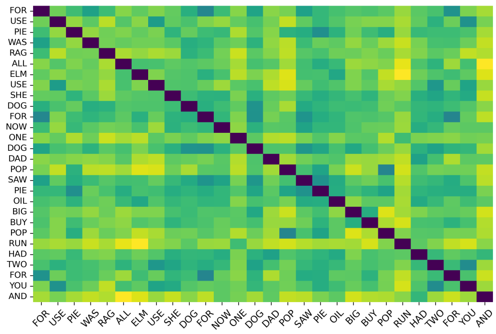
Figure 38: Similarity matrices of bottleneck representations (zoomed-in) from participant M1: Euclidean distance between bottleneck vectors representing short words.



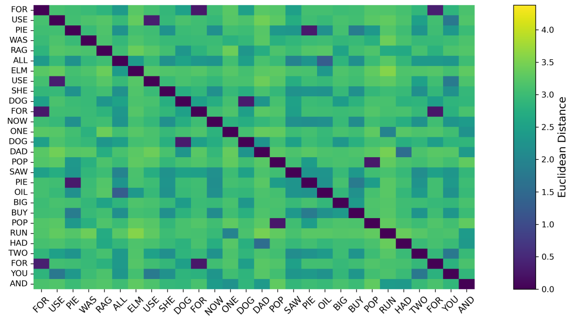
(a) Speaker-Specific M2

(b) Speaker-Invariant M2

Figure 39: Similarity matrices of bottleneck representations (zoomed-in) from participant M2: Euclidean distance between bottleneck vectors representing the same word label.

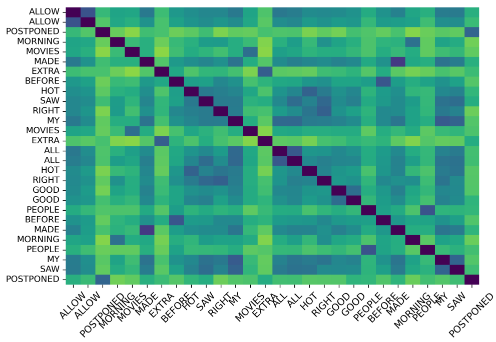


(a) Speaker-Specific M2

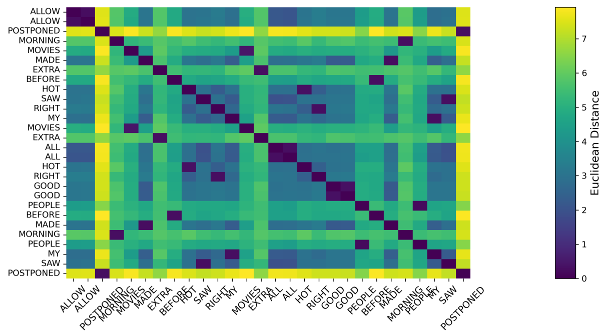


(b) Speaker-Invariant M2

Figure 40: Similarity matrices of bottleneck representations (zoomed-in) from participant M2: Euclidean distance between bottleneck vectors representing short words.

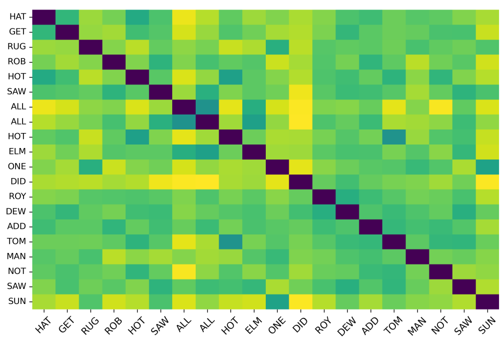


(a) Speaker-Specific M3

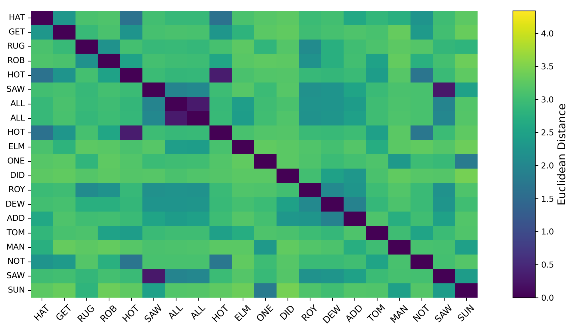


(b) Speaker-Invariant M3

Figure 41: Similarity matrices of bottleneck representations (zoomed-in) from participant M3: Euclidean distance between bottleneck vectors representing the same word label.

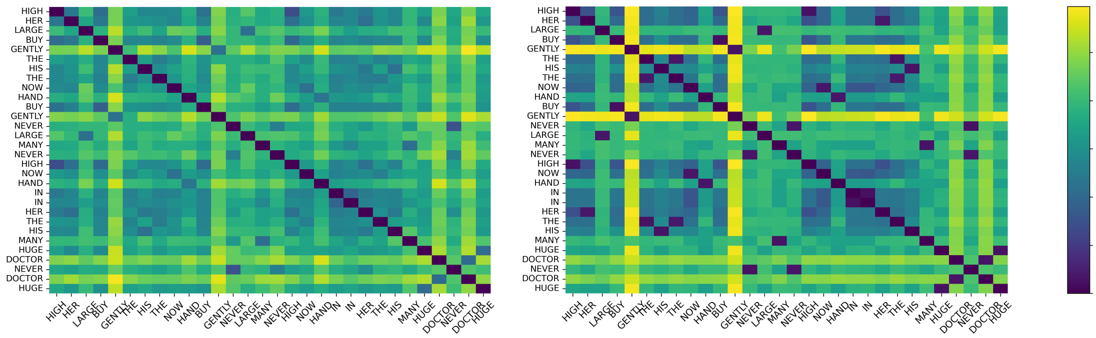


(a) Speaker-Specific M3



(b) Speaker-Invariant M3

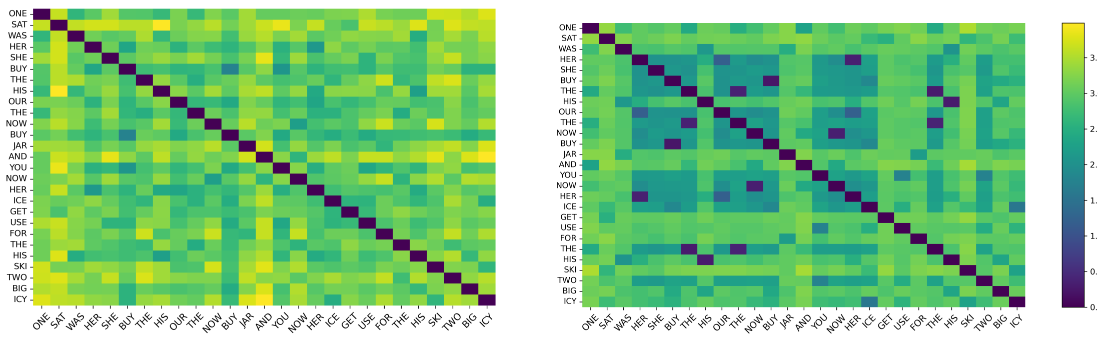
Figure 42: Similarity matrices of bottleneck representations (zoomed-in) from participant M3: Euclidean distance between bottleneck vectors representing short words.



(a) Speaker-Specific M4

(b) Speaker-Invariant M4

Figure 43: Similarity matrices of bottleneck representations (zoomed-in) from participant M4: Euclidean distance between bottleneck vectors representing the same word label.



(a) Speaker-Specific M4

(b) Speaker-Invariant M4

Figure 44: Similarity matrices of bottleneck representations (zoomed-in) from participant M4: Euclidean distance between bottleneck vectors representing short words.