Voicing Silent Speech

David Gaddy

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-68
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-68.html

May 11, 2022

Voicing Silent Speech

by

David Gaddy

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Dan Klein, Chair
Professor Gopala Anumanchipalli
Professor Keith Johnson

Spring 2022

Voicing Silent Speech

Abstract

Voicing Silent Speech

by

David Gaddy

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Dan Klein, Chair

This thesis concerns the task of turning silently mouthed words into audible speech. By using sensors that measure electrical signals from muscle movement (electromyography or EMG), it is possible to capture articulatory information from the face and neck that pertains to speech. Using these signals, we aim to train a machine learning model to generate audio in the original speaker's voice that corresponds to words that were silently mouthed. We call this task voicing silent speech.

Voicing silent speech has a wide array of potential real-world applications. For example, it could be used to allow phone or video conversations where other people around the person speaking can't hear anything they say, or it could be useful in some clinical applications for people who can't speak normally but still have use of most of their speech articulators.

There have been several papers in the past that have looked at the problem of converting EMG signals to speech. However, these prior EMG-to-speech works have focused on the artificial task of recovering audio from EMG that was recorded during normal vocalized speech. In this work, we will instead generate speech from recordings where no actual sound was produced. Models trained only on vocalized speech perform poorly when applied to silent speech due to signal differences between the two modes. Our work is the first to train a model on EMG from silent speech, allowing us to overcome these signal differences.

Training with EMG from silent speech is more challenging than with EMG from vocalized speech, because when training on vocalized EMG data we have time-aligned speech targets but when training on silent EMG data there is no simultaneous audio. Our solution is to adopt a target-transfer approach, where audio output targets are transferred from vocalized recordings to silent recordings of the same utterances. To do this cross-modal training, we need to account for the fact that the two recordings are not time-aligned, so a core component of our work concerns finding the best way to align the vocalized speech targets with the silent utterances.

To enable development on this task, we collect and release a dataset of nearly twenty hours of EMG speech recordings, nearly ten times larger than previous publicly available datasets. We then demonstrate a method for training a speech synthesis model on silent EMG and propose a range of other modeling improvements to make the synthesized outputs more intelligible. We validate our methods with both human and automatic metrics, demonstrating major improvements in intelligibility of generated outputs.

To my family

Contents

Acknowledgments

I'm very grateful for all of the many people who have helped me throughout my PhD.

First, I would like to thank Dan Klein for being such a great advisor. Dan has played a major part in making grad school a pleasant experience by always being extremely positive and supportive. I really appreciate how open he has been to letting me explore different research directions, including the subject of this thesis, even when they fall outside the range of tasks typically done in our group. His feedback on writing and presentations has always greatly improved my work, and he has always had extremely helpful advice on how to proceed in research and in my career.

I also want to thank the other professors who have helped me through the process of learning about NLP and writing my dissertation, including Gopala Anumanchipalli, John DeNero, Keith Johnson, Marti Hearst, and David Bamman. Outside of Berkeley, I've learned a lot from my internship hosts, and prior to my PhD my undergraduate mentor Regina Barzilay played a major role in my development as a researcher.

Another big thanks goes to all the other students of the Berkeley NLP and speech groups. They have been instrumental in helping me keep up-to-date on the latest developments in NLP and speech, both by sharing interesting papers from outside Berkeley and by sharing their own excellent work.

Finally, I want to thank all of my other friends and family for their support and encouragement.

Chapter 1

Background

This thesis is about voicing silent speech, where silently mouthed words are turned into an audible voice based on sensor readings captured from muscles of the face. This chapter will give some background on silent speech to set the stage for the remainder of the thesis. We first discuss in more detail what silent speech is, then talk about possible applications of silent speech technologies. Next, we talk about the input sensors we use to capture silent speech, and finally give a summary of other work on understanding it.

1.1 What is Silent Speech?

In this thesis, the term *silent speech* refers to a mode of speaking where words are mouthed while suppressing normal speech sounds like voicing and frication. This means that air is not forced through the articulators as in normal speech, but the articulators are still moved. Silent speech may not actually be completely silent, since just the movement of the lips and tongue can create some sound, but it will be much quieter than normal speech volumes.

To better define silent speech, let us compare it to alternate modes of speaking. These modes of speaking fall along a spectrum based on how pronounced the speech is. Each category can cover a range of speaking behavior and the boundaries may not always be clear, but we can roughly divide the spectrum into four different modes of interest: vocalized speech, whisper, silent speech, and subvocal speech.

Vocalized speaking is the "normal" speech mode used to communicate in

everyday situations. The source of sounds in vocalized speaking come from both voicing, caused by activation of the vocal cords, and from other restrictions of airflow in the vocal tract such as frication and bursts (Ladefoged and Johnson, 2014).

The next mode, whisper, differs from vocalized speaking by a lack of voicing (Lim, 2011). In whispered speech, air is forced through the vocal tract but the vocal cords are no longer activated. The volume of whispered speech can vary depending on how forcefully air is pushed through, but is generally loud enough to be heard by other people who are nearby. The quietest end of the whisper spectrum is sometimes referred to as non-audible murmur (Nakajima et al., 2003). Despite the name, non-audible murmur does usually have enough airflow to produce some sound, but it is generally not loud enough for others to understand and must be picked up with a special stethoscopic microphone. In non-audible murmur, the airflow is very low and may be little more than the flow resulting from normal breathing.

Silent speech is when airflow is reduced to the point where the airflow itself does not cause any sound, but the speech articulators are still moved as if speaking. Producing silent speech may require controlling breathing to keep airflow below levels that cause sound. We have observed that when people are asked to produce silent speech, they will sometimes produce a quiet whisper or non-audible murmur instead, but can usually correct to silent speech when it is brought to their attention that they are doing so.

One final mode of speaking is subvocal speech, the internal occurrence of words in the mind when reading or thinking (Jorgensen et al., 2003). In subvocal speech, the speaker does not consciously make any attempt to move the speech articulators. However, this internal speech is often accompanied by small amounts of activation of the speech muscles which the speaker is not aware of (Edfeldt, 1959). This sometimes includes perceptible movements of the lips, but can also range to more subtle muscle activation that cannot be seen visibly. In some cases, the terms silent speech and subvocal speech have been used in the place of each other, for example when Edfeldt (1959) use the term silent for behavior we call subvocal or when Meltzner et al. (2018) use both terms interchangeably for behavior we call silent speech. Our definition here is based on the most common use of the terms.

Out of this range of speaking modes, this thesis will focus on the silent speech variety to balance between privacy and the availability of necessary signals. Because silent speech intends to mime normal speech as closely as

possible without sound, it is more likely than subvocal speech to contain all the necessary signals needed to fully decode the speech. As progress is made towards decoding speech from each mode, we hope that future work can shed more light on the information available in subvocal and silent speech. Finally, we note that although our data and experiments pertain specifically to silent speech, most of our methods could easily be applied to other speaking modes.

1.2 Applications

There are many different applications where silent speech could be useful. Three broad categories of applications we will discuss here are private communication, communication with some forms of reduced speaking ability, and interaction with devices.

The first application area aims to allow people to have private conversations that can't be heard by others around them. For example, silent speech could be used for holding phone or video conversations where the people around you can't hear anything you say but the person on the other end of the line hears your voice normally. In this case, the silent speech could be translated into audio of the speaker's voice and played on the other end of the call as the speaking occurs. This could be useful for holding phone conversations in public or open-office settings with more privacy and less disruption to others without requiring physical separation like conference rooms. Since the silent speech input device does not rely on audio, it has the added advantage that noise in the environment is not picked up, making it useful for noisy environments as well. The conversation does not necessarily need to be between remote participants - it could also be useful for holding conversations between two people in close proximity where the sound is played through headphones to the other participant.

Another potential use case for silent speech could be clinical applications for people who are no longer able to produce normal audible speech but still have use of most of their speech muscles. For example, it could be beneficial for patients who have undergone a laryngectomy, where the larynx has been removed due to trauma or disease (Meltzner et al., 2017). It may also be useful for patients with some types of diseases affecting the nerves or muscles (Kapur et al., 2020). Of course, the effectiveness in these cases will depend on how much of the speech muscles are intact and on the availability of training data

of a similar character.

One final application area of silent speech is as an input for computers, phones, or other devices. Speech can be a very effective method for interaction with devices, both through the use of virtual assistants and for dictation of text. It is generally much faster than other word input methods like keyboards, and can be particularly useful for mobile devices that do not have full-sized keyboards. However, users may find vocalized speech to be inappropriate in some settings due to privacy concerns and a desire to avoid disturbing others, and silent speech could be used to alleviate these problems.

Based on the different use cases, there are two possible outputs we might want from a silent speech system. For applications where one person is talking to another, an output of synthesized audio is often ideal so that the other person can quickly understand what is being said. On the other hand, for applications where a person is talking to a device a text output may be more appropriate, similar to the output of an audio-based automatic speech recognition system. In this thesis, we will denote the tasks of outputting text and audio as recognition and voicing, respectively. Our focus will be on the task of voicing silent speech, but we will also briefly discuss some work we have done on silent speech recognition near the end.

While one could perform voicing indirectly by first doing text recognition and then synthesizing audio with a text-to-speech system, voicing the speech directly has several advantages over this approach. First, a direct voicing approach is more suited for real-time streaming, where audio is generated immediately after the corresponding speech movement, which would allow more fluid conversation with silent speech communication devices. Second, an intermediate text step could introduce unnatural errors that may be harder for a human to interpret. When a voicing system cannot correctly distinguish a sound, it may make an ambiguous output or generate a phonetically related sound, which could allow the human listener to fill in or correct based on context. When a recognition system makes an error, it is more likely to substitute an incorrect word which does not closely match the intended word, and it does not have a natural way to indicate ambiguity that a human can resolve.

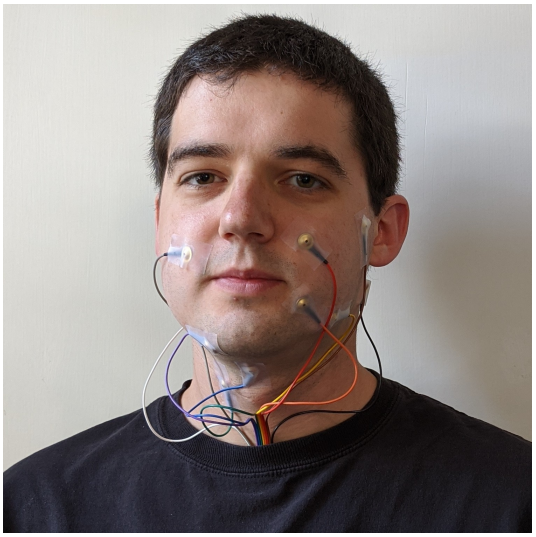

Figure 1.1: Electromyography (EMG) electrodes placed on the face can detect muscle movements from speech articulators.

1.3 Electromyography

In this work, the inputs we use to capture speech information come from surface electromyography, or EMG. Surface EMG uses electrodes placed on top of the skin to measure electrical potentials caused by nearby muscle activity.[1] By placing electrodes around the face and neck, we can capture signals from muscles that are important for speech, which may include the jaw, tongue, lips, larynx, and soft palate. Figure 1.1 shows the EMG electrodes used to capture signals.

The electrical signals captured by EMG originate from the process by which muscle cells are activated, which we will briefly describe here (Scanlon and Sanders, 2018, Chapter 7). Prior to firing, muscle cells have an electrical potential across their outer membrane from an imbalance of charged ions. When a nerve signal tells the muscle to activate, ion channels open to let ions flow into the cell, which trigger the chemical process that causes the muscle to contract. Afterwards, ion pumps move the ions back across the cell membrane

[1] Another form of EMG uses needle electrodes that are inserted into muscles, but all EMG in this work refers to the surface EMG variety, sometimes called sEMG.

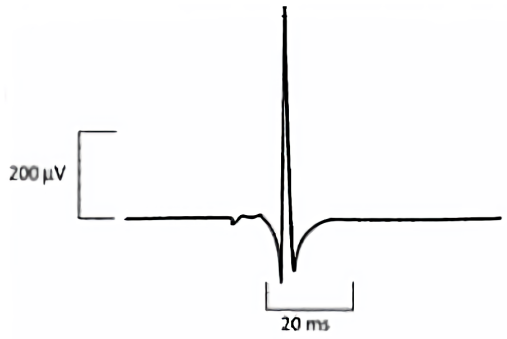

Figure 1.2: The electrical pulse from a single motor unit action potential.

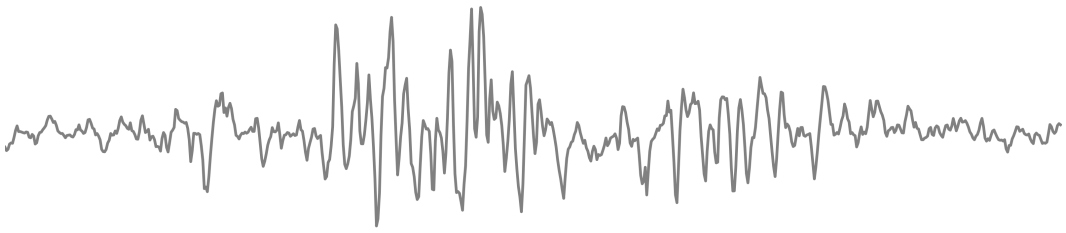

Figure 1.3: A signal captured by a single EMG channel.

to recharge. This movement of charged ions into and out of the cell causes an electrical pulse to propagate out, and these pulses are what the EMG sensors will capture.

The pulse from a single activation is shown in Figure 1.2. This pulse comes from a group of muscle cells called a motor unit which are all triggered by a single motor neuron, and the resulting pulse is called a motor unit action potential (Rodriguez-Carreno et al., 2012). The particular shape of the action potential can vary based on various properties of the cells involved. To increase the strength of a contraction, the frequency of firings for a unit can increase and more units can be activated. The signals captured by our surface electrodes are the combination of a large number of different action potentials, resulting in signals like the one shown in Figure 1.3.

When placed on different locations, EMG electrodes will tend to capture signals from different muscles that are nearby. In this work, we collect sig-

nals from eight different electrodes, as shown in Figure 1.1, resulting in eight channels of signal to use as inputs.

1.4 History of EMG for Speech

The idea of using EMG for speech has been around for some time. Some of the first work with EMG and speech was performed by Edfeldt (1959), who used electromyography for a scientific study of subvocal speech. His work showed that subvocal speech led to muscle activation of the larynx that could be measured with sub-surface electrodes, and investigated how various factors such as reading ability affected the amount of detectable subvocal speech. In the 1980's, several attempts were made to differentiate sounds from electromyography during vocalized speech using simple data-analysis and pattern-matching techniques. Sugie and Tsunoda (1985) and Morse and O'Brien (1986) used EMG to discriminate between small sets of sounds, with classification accuracies ranging from 35% for 17 word sets to 97% for two word sets.

In the early 2000's, people began applying automatic speech recognition systems based on hidden Markov models to EMG-based speech, including on silent and subvocal speech. Initially these works operated over vocabularies of approximately 10 words (Chan et al., 2002; Jorgensen et al., 2003; Maier-Hein et al., 2005), but over time vocabulary size improved to recognize approximately 100 words with word error rates as low as 10-20% (Jou et al., 2006; Schultz and Wand, 2010; Wand and Schultz, 2011). Since then, other works have continued to study the task of recognizing text from EMG (Wand et al., 2014b; Meltzner et al., 2018; Kapur et al., 2018), but these have still been limited to fairly restricted vocabulary sizes. While recognizing text from EMG is not the primary task we will focus on in this thesis, Chapter 6 will discuss some of our work on recognizing text, where we move to a large open vocabulary and apply some of the more recent techniques from the speech recognition literature.

In addition to the recognition (speech-to-text) work described so far, there were also several attempts to convert EMG speech to audio prior to our work. Toth et al. (2009) considered the use of voice conversion tools based on Gaussian mixture models (GMM) for EMG-to-speech conversion. They trained their system on vocalized speech and tested on both vocalized and silent speech, finding results on vocalized speech to be fairly intelligible but on

silent speech to be mostly unintelligible. Note that while testing on vocalized speech is useful for evaluating progress, it represents an artificial setup since the speech already has audio and so does not need for it to be reconstructed from EMG. Diener et al. (2015) introduced the use of neural networks for EMG-to-speech synthesis, showing substantial improvements over the GMM-based approach, and later work by many of the same authors continued to improve models for this task (Janke and Diener, 2017; Diener et al., 2018). However, these works focused just on synthesis from vocalized speech, training and testing only on that speech mode. Our work is the first to instead train on silent speech signals, allowing us to achieve more intelligible speech synthesis from silent EMG.

1.5 Alterative Input Sensors

While the focus of this work will be on using EMG as an input for silent speech, it is not the only possible way to capture speech that does not rely on audio. Several alternatives include visual inputs, electromagnetic articulography, ultrasound, or brain signals from EEG, ECoG, or fMRI. We will briefly discuss some of the tradeoffs of these inputs.

One alternative method for capturing silent speech is to use video to visually read the lips of speakers (Petridis and Pantic, 2016; Chung et al., 2017; Shi et al., 2022). A recent model from Shi et al. (2022) using this input achieved a word error rate of 27% in a multi-speaker setting. Some advantages of this method are the ease of capturing inputs and the large amount of speaking video data available that can be used for training. One potential downside is that only the outside of the face can be seen, which may mean important information is missing. While humans may be able to perform lip reading well when given sufficient contextual clues, some tests have shown that even professional human lip readers have error rates of over 70% from visual signals alone (Chung et al., 2017). There are also trade-offs for ease of use, since EMG may be more convenient when walking around but video more convenient when sitting in front of a computer.

Another possible sensor for silent speech input is electromagnetic articulography, or EMA, which uses magnets attached to the lips and tongue to track their movement (Wrench and Richmond, 2000). While EMA has very accurate information about movement of the speech articulators, making it

great for use in the lab, its need for attaching items to the tongue makes it too invasive for many uses as an everyday communication device.

Ultrasound imaging of the inside of the mouth is another possible input for capturing silent speech (Hueber et al., 2010; Kimura et al., 2019). Ultrasound has the advantage of being able to see the tongue without placing sensors inside the mouth. It may be less effective at capturing the lips and so is sometimes combined with visual inputs to capture that information (Hueber et al., 2010). Another downside is that current ultrasound sensors are often be more expensive and bulky than EMG or video.

Finally, there are several possible inputs based on reading signals from the brain. For example, EEG sensors can read electrical signals from the brain off the surface of the skin, just as EMG sensors do for muscles (D'Zmura et al., 2009). However, due to signal attenuation by the skull, these sensors may have too low a resolution to capture enough information for decoding speech. ECoG sensors implanted inside the skull can capture more fine-grained information (Anumanchipalli et al., 2019), though they require surgery to implant. Imaging techiques such as fMRI can also be used to capture speech information from the brain (Price, 2012), but the large size and cost of these machines make them impractical for many use cases.

Chapter 2

Data Collection

To enable our work on voicing silent speech, we collect and release a dataset of EMG signals and time-aligned audio during both silent and vocalized speech. The dataset contains nearly 20 hours of facial EMG signals from a single speaker. To our knowledge, the largest public EMG-speech dataset previously available contains just 2 hours of data (Wand et al., 2014a), and many papers continue to use private datasets for EMG-speech tasks. We hope that our public release will encourage development on these tasks and allow for fair comparisons between methods.

In this chapter, we first describe the equipment used to record signals and then the structure of our dataset.

2.1 Recording Equipment and Setup

To record EMG signals we use the Cyton Biosensing Board (OpenBCI, 2014), a device designed for measurement of biopotentials – the small voltages generated by the body. The primary electrical function of the board is to amplify voltages and convert them to digital values with an analog-to-digital converter (ADC). These values are then streamed to a computer over Wi-Fi, where they are saved. With an amplifier gain of 24 and an 24-bit ADC, the board is theoretically capable of measuring voltages with a resolution of approximately .01 microvolts, though the electrical components have a typical noise level of .3 microvolts, making that the practical resolution (Texas Instruments, 2012). The board is capable of recording eight channels simultaneously, and we record

at a rate of 1000 samples per second.

The voltage measurements of the Cyton board are made with respect to a reference electrode placed at a different location. In our data we use a single reference electrode that is shared across all recording channels, which is known as a monopolar configuration (Jamal, 2012). We place this reference behind one ear. When each channel uses it's own reference electrode it is known as a bipolar configuration, and two electrodes are needed for each channel. We choose a monopolar configuration over a bipolar one to reduce the number of electrodes that need to be placed and because a bipolar configuration requires more care in choosing locations to get the right signals.

The Cyton board also has circuitry to help maintain a steady base voltage across electrodes by using an additional "bias" electrode where the voltage is driven by the board. This base voltage is known as the common mode voltage, since it appears across all measurement electrodes. Reducing the common mode voltage can help reduce some types of noise in the measurements. The circuitry used to cancel the common mode voltage is sometimes called a driven right leg circuit, since for measurement of electrocardiography (ECG) signals the driven electrode is often placed on the leg. We place a driven electrode behind the opposite ear as our reference electrode.

The electrodes themselves are simply a metal cup attached to the data collection board with a wire, as shown in Figure 2.1. The surface of the electrodes are gold plated, which can affect signal quality based on how the surface material interacts chemically and electrically with the skin. Ten20 conductive electrode paste is placed within the cup of the electrodes prior to placement. This paste helps create a better electrical connection between the electrodes and the skin. Although not explored in this work, dry electrodes that do not require electrode paste are also possible with additional circuitry.[1] In some cases, this additional circuitry may be placed on or near the electrodes themselves, in which case they are called active electrodes. In contrast, the electrodes we use are called passive electrodes.

The electrodes are individually attached to the face and neck with tape. The locations where we place the electrodes are described in Table 2.1, and can also be seen visually in Figure 1.1. These locations were chosen to be close to many of the primary speech articulators and to have some similarity

[1] One important property for dry electrode circuitry is an amplifier with high input impedance.

Figure 2.1: Gold-plated cup electrodes used for capturing EMG.

to locations used by prior work such as Wand et al. (2014a) and Kapur et al. (2018). We did not carefully optimize electrode locations and expect there is room for improvement in these location choices.

The EMG signals we aim to capture can have magnitudes up to several millivolts, but the intensity can also be much lower depending on the strength of muscle activation and distance from the electrodes. When attached to a speaker at rest, we typically observe 10 to 20 microvolts of background signal.

Before using the collected signals, several filters are used to reduce noise and normalize the signals. First, a series of IIR notch filters at integer multiples of 60 Hz are used to reduce noise from AC electrical mains, which shows up significantly in the signals prior to filtering, often with magnitudes higher than the signals we aim to capture. Next, a high pass Butterworth filter with cutoff 2 Hz is used to remove constant offset and slow-moving drift from the collected signals, since offsets can vary widely and do not generally contain useful information. For both filters, forward-backward filtering is used to avoid phase shifts.

Audio is recorded from a built-in laptop microphone at 16 kHz. Prior to use, background noise is reduced in the audio using a spectral gating al-

	Location
1	left cheek just above mouth
2	left corner of chin
3	below chin back 3 cm
4	throat 3 cm left from Adam's apple
5	mid-jaw right
6	right cheek just below mouth
7	right cheek 2 cm from nose
8	back of right cheek, 4 cm in front of ear
ref	below left ear
bias	below right ear

Table 2.1: Electrode locations.

gorithm,[2] which uses a sample of silence recorded at the beginning of each session to identify thresholds below which inputs should be considered noise across different frequency bands. We also normalize the volume of the audio across different examples so that the peak volume of each example is approximately constant. We measure volume with the root-mean-square value of 30 ms windows of sound, take a max over all windows in an example, smooth these peak values across nearby examples, and then scale each example's volume based on these values.

2.2 Dataset Structure

Our data comes from a single male speaker, the author of this work. All models in this thesis will focus on a single-speaker setup using this data, and so will only need to understand EMG from that speaker and output a single voice. The vast majority of other work on decoding EMG speech has also used a similar speaker-dependent setup, though Wand and Schultz (2009) did perform some cross-speaker experiments for EMG speech recognition. Extending the task of voicing silent speech to multiple speakers is an important direction for future work.

[2]https://pypi.org/project/noisereduce/

Recording of the dataset was broken down into sessions of approximately one hour in length. Electrodes were reattached for each session and so may have minor changes in position between different sessions. Within a session the electrodes remained in place, though signal differences could still occur as electrodes became more settled or more loose over time. Each session was recorded on a different day to avoid strain from excessive talking.

During data collection, text prompts consisting of a single sentence to be read were displayed on a computer screen at a time. After reading the sentence, the subject pressed a key to advance to the next sentence. If they were unhappy with a recording, they could press another key to re-record an utterance. A real-time display of EMG signals was used to monitor for any excessive noise during recording. Such noise often comes from loose electrodes and could generally be corrected by pressing the electrodes back into contact.

The primary portion of the dataset consists of parallel silent and vocalized data, where the same utterances are recorded using both speaking modes. This parallel data is used by our methods for training on silent EMG, which will be discussed in a later chapter. The parallel examples can be viewed as tuples (E_S, E_V, A_V, T) of silent EMG, vocalized EMG, vocalized audio, and the text prompt, where E_V and A_V are time-aligned. Although we also recorded audio during the silent speech, these recordings generally do not contain useful information and are ignored. Figure 2.2 shows an example from the parallel data collected. Both speaking modes of an utterance were collected within a single session to ensure that electrode placement is consistent between them. First the full set of vocalized utterances for the session were collected, then the same utterances were recorded with silent speaking.

Another set of sessions in the dataset contain only vocalized speaking. We refer to these instances as non-parallel data, and represent them with the tuple (E_V, A_V, T).

2.3 Domains

For comparison, we recorded data from two domains: one with an open vocabulary and one with a closed vocabulary. We describe each of these domains below.

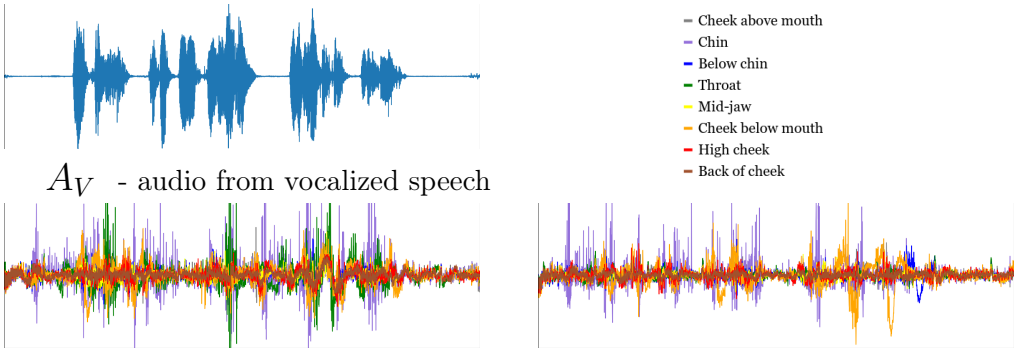

Figure 2.2: An example of signals collected during a parallel session. The vocalized speech signals A_V and E_V are collected simultaneously and so are time-aligned, while the silent signal E_S is a separate recording of the same utterance without vocalization. Colors represent different electrodes in the EMG data. Not pictured, but also included in our data are the utterance texts, in this case: "It is possible that the infusoria under the microscope do the same." (from H.G. Well's *The War of the Worlds*).

2.3.1 Open-Vocabulary Condition

The majority of our data was collected with sentences read from books. This data is open-vocabulary, meaning we do not restrict the vocabulary in any way, and the development and test sets contain words that were never seen in the training set. The text that was read came from two public domain books from Project Gutenberg: *The Adventures of Sherlock Holmes* by Arthur Conan Doyle and *The War of the Worlds* by H. G. Wells.[3] We collected 17 total sessions in this condition: 7 sessions with parallel silent and vocalized utterances, and 10 non-parallel sessions with only vocalized utterances. A summary of dataset features is shown in Table 2.2. We select a validation and test set randomly from the silent parallel EMG data, with 200 and 100 utterances respectively. Note that during testing, we use only the silent EMG recordings E_S, so the vocalized recordings of the test utterances are unused.

[3]https://www.gutenberg.org/

Open-Vocabulary Condition
Parallel Silent / Vocalized Speech
(E_S, E_V, A_V, T)
3.6 hours silent / 3.9 hours vocalized
Average session has 30 min. of each mode
1588 utterances
Non-parallel Vocalized Speech
(E_V, A_V, T)
11.2 hours
Average session length 67 minutes
5477 utterances
Total
18.6 hours
Average of 16 words per utterance
9828 words in vocabulary

Table 2.2: Open-vocabulary data summary

2.3.2 Closed-Vocabulary Condition

In our second data condition, we use a restricted domain with words from a closed, or limited, vocabulary. Restricting the vocabulary makes it easier for our models to generate clear audio because it reduces the possible outputs a model must choose from. Even if the model can't distinguish every sound individually, with a restricted vocabulary it may be able to identify the rest of the word and use that to determine what to output. In addition, using a closed vocabulary reduces the number of different contexts where each phoneme could occur, which may make identification easier if phonemes appear differently in different contexts as they do in audio-based speech.

To create a closed-vocabulary data condition, we generate a set of date and time expressions for reading. These expressions come from a small set of templates such as "`<day-of-week> <month> <day-of-month>`" which are filled in with randomly selected values. Over 50,000 unique utterances are possible from this scheme. We collected a single session of this date and time data, and Table 2.3 summarizes the properties of the data collected in this

Closed-Vocabulary Condition
Parallel Silent / Vocalized Speech
(E_S, E_V, A_V, T)
26 minutes silent / 30 minutes vocalized
Single session
500 utterances
Average of 4 words per utterance
67 words in vocabulary

Table 2.3: Closed-vocabulary data summary

condition. A validation set of 30 utterances and a test set of 100 utterances are selected randomly, leaving 370 utterances for training.

Chapter 3

Methods

This chapter will discuss our methods for the task of voicing silent speech, where we aim to translate EMG signals into speech audio.

Instead of training a model that goes directly from EMG to audio waveforms, we break down the model into two main parts. First, we predict a set of audio features with a model that we will call the transduction model, then we use a separate vocoder model to turn those audio features into a raw waveform. This breakdown is commonly used in the speech synthesis literature (Shen et al., 2018; Ping et al., 2018; Li et al., 2019) and is often more effective since it allows the initial model to focus on the high-level structure of the sound while letting the vocoder handle the details of generating waveforms.

Conceptually, our neural transduction model will translate from a sequence of EMG signals E to a time-aligned sequence of audio A. By time-aligned, we mean that for every frame (small chunk) of input EMG, we will output a frame of audio that directly corresponds to the sound being expressed in the EMG at that point in time. The initial components of our model extract features for the EMG at each frame, so there is a one-to-one correspondence between EMG feature frames and audio feature frames on the input and output. Figure 3.1 illustrates this correspondence. Using a time-aligned model structure has the advantages of giving the model more inductive bias and allowing real-time streaming into audio with only a few minor extensions, as discussed later in Section 4.3.4.

When training solely on *vocalized* EMG data (as was done by prior work) training the model is straightforward, because for every frame of EMG the simultaneously recorded audio gives us a target output to regress towards.

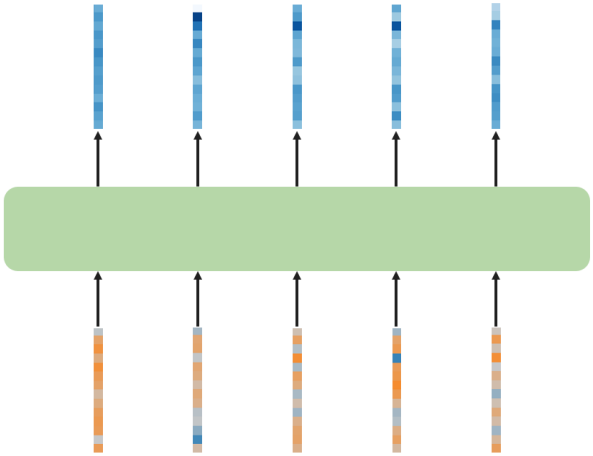

Figure 3.1: The core of our neural transduction model goes from a sequence of EMG features (bottom) to a sequence of time-aligned audio features (top).

However, our experiments in Chapter 4 show that training on vocalized EMG alone leads to poor performance when testing on silent EMG because of differences between the two speaking modes. A core contribution of our work is a method for training the transduction model on silent EMG signals, which no longer have time-aligned audio to use as training targets. Using the set of parallel utterances that we recorded in both silent and vocalized speaking modes, we find alignments between the two recordings and use them to associate speech features from the vocalized instance with the silent EMG frames.

Our work also introduces several other modeling improvements compared to prior work. We improve the EMG feature extraction by using learned rather than manual features, the model architecture by using Transformers in the place of LSTMs, and the learning signal by adding an auxiliary loss to predict phoneme labels.

We will break down the discussion of our model into several parts. First we will describe how features can be extracted from the raw EMG signals, and we'll describe the core transduction model which translates EMG features to audio features. Next, we'll introduce our methods for training on silent speech by aligning to a parallel vocalized utterance and describe our auxiliary phoneme loss. Finally, we'll discuss the vocoder model which synthesizes the

final audio waveforms from the audio features.

3.1 Input Features

The first step of our model is to turn the raw EMG into features. The features will operate at a more coarse temporal level than the raw signals, and each feature vector will summarize a small section of the EMG input. The stride of the EMG feature frames is chosen to be the same as the stride used by the audio features output by the transduction model so that inputs and outputs line up. A stride of 11.6 ms is used to be compatible with the audio features used by many recent vocoders.

Prior to feature extraction, AC electrical noise is removed from the EMG signals using band stop filters at harmonics of 60 Hz, and DC offset and drift are removed with a 2 Hz high-pass filter. We also perform soft de-spiking to remove very large values by feeding the input through a scaled tanh function ($\nu \tanh \frac{x}{\nu}$) with the maximum scale ν set to 1 mV.

Our work has explored two different methods for extracting features. First we consider manually defined features like those used in prior EMG-speech work, and then try a new feature extraction method based on learning features from raw EMG signals. We describe each of these two feature extraction methods in more detail below.

3.1.1 Manual Features

For our manual feature set, the primary features we use are the time domain features from Jou et al. (2006), which are commonly used in the past EMG-speech literature (Schultz and Wand, 2010; Diener et al., 2015). After splitting the signal from each channel into low and high-frequency components (x_{low} and x_{high}) using a triangular filter with cutoff 115 Hz, the signal is windowed with a frame length of 31 ms and stride of 11.6 ms. For each frame, a set of five features are used describe major properties of the signal, as follows:

$$\left[\frac{1}{n}\sum_i (x_{low}[i])^2,\ \frac{1}{n}\sum_i x_{low}[i],\ \frac{1}{n}\sum_i (x_{high}[i])^2,\right.$$
$$\left.\frac{1}{n}\sum_i |x_{high}[i]|,\ \text{ZCR}(x_{high})\right]$$

where ZCR is the zero-crossing rate, the number of times the signal's sign changes from one sample to the next. To implement these features in a similar manner as prior work, we first re-sample our EMG to 516.8 Hz. This is scaled from the 600 Hz used by Jou et al. (2006) because we use a different frame stride of 11.6 instead of 10 ms. We perform low-pass filtering for x_{low} with two passes of averaging with windows of 9 values, and then subtract this from the original signal to get x_{high}. The features are then calculated over 16-sample windows with a stride of 6.

Some other work on EMG processing has used frequency-domain features (Kapur et al., 2018) based on a short-time Fourier transform (STFT), where small windows of signal are converted to the frequency domain. In our initial exploration, these were not as effective as the above time-domain features when used alone, but did provide some additional improvement when used in combination. For each 31 ms window with 16 samples, we calculate a 16-point STFT to append to our feature set, which gives us 9 additional features.

The two representations result in a total of 112 features to represent the 8 EMG channels. The features are normalized to approximately zero mean and unit variance before use.

3.1.2 Learned Feature Extraction

As an alternative to using manual features like in prior work, our work also investigated whether features could be learned from raw EMG signals with minimal preprocessing. To learn the features, we add a set of convolutional neural network layers to the beginning of the transduction model to act as feature extractors. These layers are trained along with the rest of the transduction model, and give the model the ability to learn its own features. This variant follows recent work in speech processing from raw waveforms like Collobert et al. (2016) and Schneider et al. (2019).

Our convolutional architecture uses a stack of 3 residual convolution blocks inspired by ResNet (He et al., 2016), but modified to use 1-dimensional convolutions. The architecture used for each convolution block is shown in Figure 3.2. Along the primary computation path, two convolution layers with a width of 3 over the sequence are used with a rectified linear activation (ReLU) in between, and along the other "shortcut" path a single width-1 convolution (linear transformation with no sequence aggregation) is performed. The final output of the block is then the sum of the two path outputs followed by a

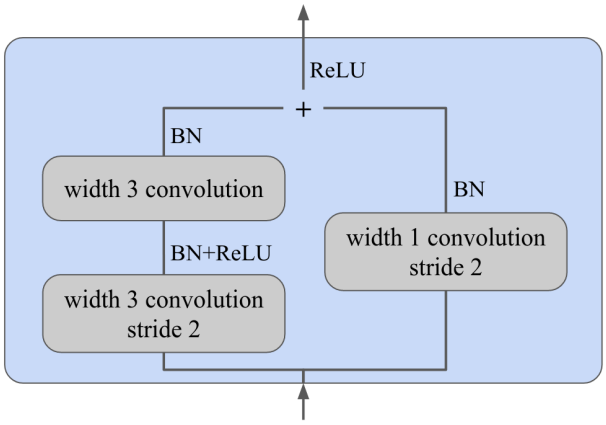

Figure 3.2: The convolution block architecture used to extract learned features from raw EMG signals.

ReLU activation. Each convolution is followed by a batch normalization operation (Ioffe and Szegedy, 2015) (BN in figure). The strides at the beginning of the block are set to 2, so that each block downsamples by 2 for a total length reduction of 8 over the three layers. EMG signals are resampled from 1000 Hz to 689 Hz before being fed into the convolution layers, so that after this downsampling the stride of frames will be 11.6 ms. All convolutions have channel dimension 768.

Before feeding EMG signals into the convolution layers, we re-scale them so that a unit value corresponds to 20 µV. During training, we randomly shift the EMG signals by up to 8 samples, so that the convolutional layers will see slightly different views of the inputs.

3.2 EMG to Speech Feature Transduction

After features are extracted, the transduction model is the component that translates the sequence of EMG features into a sequence of audio features. We will denote the featurized version of the signals used by the transduction model $E'_{S/V}$ and A'_V for EMG and audio respectively. Both feature types use the same frame rate, so when signals are time-aligned the frames $E'[i]$ and $A'[i]$ will match up.

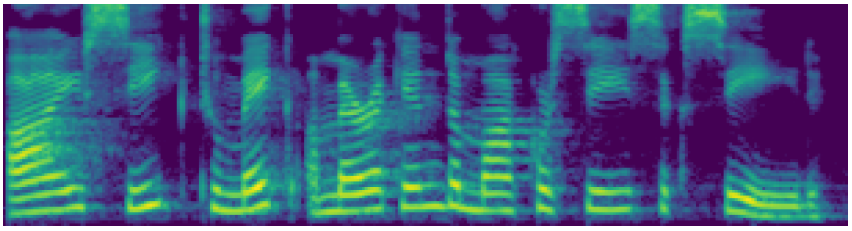

Figure 3.3: A mel-spectrogram representation of audio.

The EMG features that are input to the transduction model are projected up to the model dimension with a linear layer and fed into the main model layers at every time-step. We also explored including an embedding of the session index with each time-step to allow the model to account for differences in electrode placement, but this embedding did not substantially improve performance so was not used in our final experiments.

For audio feature outputs, we predict an 80-band mel-spectrogram. A mel-spectrogram is a frequency-domain representation where a STFT output is bucketed along the frequency dimension with 80 triangular windows scaled according to the Mel scale, which uses larger windows for higher frequencies. Figure 3.3 illustrates a mel-spectrogram for an audio sample of a voice. The spectrogram parameters were chosen to match those of our vocoder: sample rate 22050 (audio is resampled to match this rate), FFT and window size 1024, and hop 256. In earlier versions of our model, we also explored using Mel-frequency cepstral coefficients (MFCCs) as our audio features and were able to achieve similar results with those as well. The output features were scaled to have a mean of zero and a standard deviation of 0.25. We found that this scaling improved stability of the model compared to larger ranges.

Our training loss for the transduction model is the Euclidean distance between the predicted mel-spectrogram features and the aligned target features at each time-step. We use this as our loss to match the distance used for alignment in Section 3.3.2 below and because we found it to work well empirically. Compared to a mean squared error loss, our loss puts less emphasis on large prediction errors. This property makes it more similar to a loss based on ℓ_1 distance, which has been used by other work with spectrogram predictions like Wang et al. (2017).

We explored two different neural architectures for our transduction model:

LSTMs and Transformers. Both of these architectures propagate information across time, but do so in different ways. LSTMs have been explored in past work for EMG-to-speech by Janke and Diener (2017), but we are the first to explore the use of Transformers for this task. We describe each neural architecture below.

3.2.1 LSTM

One possible architecture for the transduction model is a recurrent Long Short-term Memory network, or LSTM (Hochreiter and Schmidhuber, 1997). An LSTM is a type of recurrent neural network, meaning it passes information one step at a time across adjacent positions from time-step $t-1$ to t. To help remember information across longer horizons, LSTMs use a cell state to pass information and a set of gates to control how information flows into and out of the cell at each position. The equations of the LSTM with input x_t and output h_t are as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Our model uses a bidirectional LSTM, allowing information to propagate forwards and backwards in time by using one LSTM in each direction for each layer of the model (Schuster and Paliwal, 1997). Based on validation-set tuning, we use a model with 3 bidirectional LSTM layers of 1024 hidden units, followed by a linear projection to the speech feature dimension. Dropout 0.5 is used between all LSTM layers (Srivastava et al., 2014), as well as before the first layer and after the last LSTM layer.

3.2.2 Transformer

As an alternative to the LSTM, we also explore using the more recent Transformer architecture (Vaswani et al., 2017), which can access distant information more directly through the use of an attention mechanism. In the Transformer architecture, each layer of the model consists of a self-attention mech-

anism followed by a feed-forward sub-layer, where self-attention aggregates information across the sequence and feed-forward sub-layers process information at each position.

The self-attention mechanism is made up of multiple attention heads, where each head attends to other positions in the sequence, choosing which locations to pull information from. In particular, for each position i of the sequence, an attention head forms a weighted sum of values across all other positions j. Weights are computed using a softmax over scaled dot products from projections of the input x:

$$a_{ij} = \text{softmax}\left(\frac{(W_K x_j)^\top (W_Q x_i)}{\sqrt{d}}\right)$$

where W_K and W_Q are learned matrices that project down to dimension d and the softmax is over positions j. Values to be aggregated come from a projection to dimension d by another learned matrix W_V, resulting in an output

$$h_i = \sum_j a_{ij} \left(W_V x_j\right)$$

The feed-forward component of each layer consists of an up-projection, non-linearity, then down-projection back to the original size. Both the attention and feed-forward sub-layers are separately wrapped with a residual connection and normalization: $\text{layernorm}\left(x + \text{sublayer}(x)\right)$

We use six of these Transformer layers, with 8 heads, model dimension 768, query dimension d of 96, and feed-forward dimension 3072. Dropout 0.2 is used after the feed-forward non-linearity and on the attention values. The output of the last Transformer layer is passed through a final linear projection down to 80 dimensions to give the audio feature predictions output by the model.

To capture the time-invariant nature of our task, we encode position using relative position embeddings as described by Shaw et al. (2018) rather than the more common absolute position embeddings. In this variant, a learned vector p that depends on the relative distance between the query and key positions is added to the key vectors when computing attention weights. Thus, the attention weights are computed with

$$a_{ij} = \text{softmax}\left(\frac{(W_K x_j + p_{ij})^\top (W_Q x_i)}{\sqrt{d}}\right)$$

where p_{ij} is an embedding lookup with index $i - j$, up to a maximum distance k in each direction. For our model we use $k = 100$, giving each layer approximately 1 second of view in each direction, and set all attention weights with distance greater than k to zero.

3.3 Training on Silent Speech

To train the EMG-to-speech feature transduction model, we need speech features that are time-aligned with the model outputs to use as training targets. However, when training with EMG from silent speech, simultaneously-collected audio recordings do not have any audible speech to use as targets. In this section we describe how parallel utterances, as described in Section 2.2, can be used to associate audio feature labels from a vocalized recording with a silent one. More concretely, given a tuple $(E'_V, A'_V, E'_S, \hat{A}'_S)$ of features from vocalized speech EMG, vocalized speech audio, silent speech EMG, and predicted silent speech audio, we estimate a warped set of predicted audio features $\tilde{A}'_S$ from $\hat{A}'_S$ that time-align with A'_V.

We consider several different ways of performing the alignment by using different features to align: EMG features, CCA-projected EMG features, or audio features. We will start by describing the alignment with EMG features, which is the simplest method.

All of our alignments will make use of dynamic time warping (DTW) (Rabiner and Juang, 1993), a dynamic programming algorithm for finding a minimum-cost monotonic alignment between two sequences s_1 and s_2. DTW builds a table $d[i, j]$ of the minimum cost of alignment between the first i items in s_1 and the first j items in s_2. The recursive step used to fill this table is

$$d[i, j] = \delta[i, j] + \min\left(d[i-1, j], d[i, j-1], d[i-1, j-1]\right)$$

where $\delta[i, j]$ is the local cost of aligning $s_1[i]$ with $s_2[j]$. After the dynamic program, we can follow backpointers through the table to find a path of (i, j) pairs representing an alignment.

For our silent training, we apply DTW as described above between the vocalized and silent sequences for each example. The different alignment variants we explore vary by how they define the cost δ used by DTW. For our EMG-based alignment, this cost is simply the Euclidean distance between EMG

feature vectors:

$$\delta_{\text{EMG}}[i, j] = \|E'_S[i] - E'_V[j]\|$$

When using EMG features for alignment we use the manual features described in Section 3.1.1, even if learned features are used for the model itself.

Although the path of (i, j) pairs returned by DTW is monotonic, a single position i in the vocalized sequence may repeat several times with increasing values of j in the silent one (and vice-versa). We take the first pair from any such sequence to form a mapping $a_{VS}[i] \rightarrow j$ from every position i in E'_V and A'_V to a position j in E'_S. Using this alignment, we can create a warped sequence of predicted audio features $\tilde{A}'_S$ that aligns with A'_V using $\tilde{A}'_S[i] = \hat{A}'_S[a_{VS}[i]]$. During training of the EMG to audio transduction model, we can then compare $\tilde{A}'_S$ to A'_V when calculating a loss.[1] Figure 3.4 illustrates how an alignment is used to associate each vocalized frame with a corresponding silent frame.

In addition to training the transduction model on silent examples using this alignment, we find that also training on the vocalized signals (E'_V to A'_V) improves performance. Since the EMG and audio targets are recorded simultaneously for these vocalized examples, we can calculate the loss directly without any dynamic time warping. Each training batch contains samples from both modes mixed together. For the open vocabulary setting, the full set of examples to sample from has 3 sources: (E_S, A_V) from parallel utterances, (E_V, A_V) from the vocalized recording of the parallel utterances, and (E_V, A_V) from the non-parallel vocalized recordings.

3.3.1 CCA

While directly aligning EMG features E'_S and E'_V can give us a rough alignment between the signals, doing so ignores the differences between the two signals that lead us to want to train on the silent signals in the first place. To better capture correspondences between the signals, we use canonical correlation analysis (CCA) (Hotelling, 1936) to find components of the two signals which are more highly correlated. Given a number of paired vectors (v_1, v_2),

[1] We also tried a variant of the alignment described here where the directionality is reversed, choosing the best vocalized frame for every silent frame instead of vice-versa. Choosing the best silent frame for each vocalized performed slightly better in our experiments, perhaps because it ensures no target sounds are skipped over in the alignment.

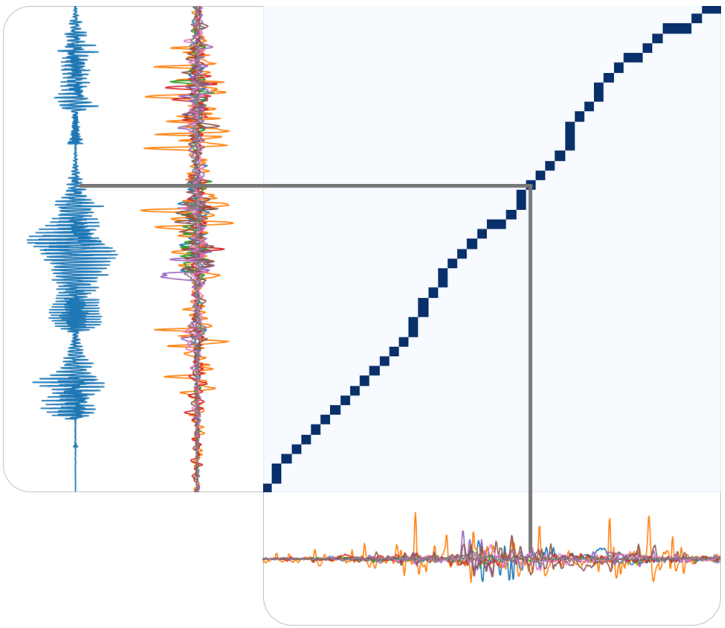

Figure 3.4: An illustration of how alignment between EMG sequences is used to find vocalized audio targets to use for training. The dark blue line represents an alignment between the two sequences. Raw EMG signals are used here for illustration, but EMG comparisons are done in the EMG manual feature space.

CCA finds linear projections P_1 and P_2 that maximize correlation between corresponding dimensions of $P_1 v_1$ and $P_2 v_2$.

To get the initial pairings required by CCA, we use alignments found by DTW with the raw EMG feature distance δ_{EMG}. We aggregate aligned E'_S and E'_V features over the entire dataset and feed these to a CCA algorithm to get projections P_S and P_V. CCA allows us to choose the dimensionality of the space we are projecting to, and we use 15 dimensions for all experiments based on validation-set tuning. Using the projections from CCA, we define a new cost for DTW

$$\delta_{\text{CCA}}[i, j] = \|P_S E'_S[i] - P_V E'_V[j]\|$$

Our use of CCA for DTW is similar to Zhou and Torre (2009), which combined the two methods for use in aligning human pose data, but we found

their iterative approach did not improve performance compared to a single application of CCA in our setting.

3.3.2 Alignment with Predicted Audio

So far, our alignments between the silent and vocalized recordings have relied on distances between EMG features. In this section, we will use audio features instead. Although the silent recordings have no useful audio signal, once we start to train a transduction model from E'_S to audio features, we can align using the *predicted* audio features $\hat{A}'_S$. Our alignment will then be between predicted features $\hat{A}'_S$ and vocalized audio features A'_V.

Training with predicted audio alignment works as follows: For each batch of training, we first run a forward pass of the model to get audio feature predictions $\hat{A}'_S$. Next, we run DTW with cost

$$\delta_{\text{audio}}[i, j] = \left\| \hat{A}'_S[i] - A'_V[j] \right\|$$

to align predictions with target audio feaures. We then treat those alignments as fixed and backpropagate errors from features paired by the alignment, just as we do for EMG. Because our training loss is also an ℓ_2 distance in the audio feature space, this alignment has the appealing property that the same metric δ is used for both the alignment and loss.

As training progresses and our predictions improve, our alignments will also improve, giving the model better learning signal. Training on the vocalized examples helps to bootstrap the process, since those examples already have aligned outputs, and we train on the two speaking modes simultaneously.

This audio-based alignment could also be mixed together with the EMG-based alignment using a weighted combination like $\delta_{\text{CCA+audio}}[i, j] = \delta_{\text{CCA}}[i, j] + \gamma\delta_{\text{audio}}[i, j]$, where γ is a hyperparameter to control the relative weight of the two terms. However, this combined alignment variant did not outperform the individual alignment variants in our experiments so is not included in our results.

3.3.3 Other Training Information

To perform batching across sequences of different lengths during training, we concatenate a batch of EMG signals across time then slice (reshape) them into

a batch of fixed-length sequences before feeding into the network. Thus if the fixed batch-sequence-length is l, the sum of sample lengths across the batch is N_S, and the signal has c channels, we reshape the inputs to size $(\lceil N_S/l \rceil, l, c)$ after zero-padding the concatenated signal to a multiple of l. After running the network to get predicted audio features, we do the reverse of this process to get a set of variable-length sequences to feed into the alignment and loss described above. When using raw EMG inputs (§ 3.1.2), the input sequence length l will be 8 times the output sequence length to account for the downsampling from raw signals to frames. We use a sequence length l of approximately 2 seconds ($l_{raw} = 1600$, $l_{frame} = 200$) and select batches dynamically up to a total length of 256 seconds. This batching strategy has the advantage of being compute-efficient since it requires minimal amounts of padding and keeps sequences short for the Transformer's l^2 compute scaling. It can also act as a form of regularization, since some context will not be available after slicing and data samples will be sliced in different ways each time they appear in a batch.

We train our model for 80 epochs using the AdamW optimizer (Loshchilov and Hutter, 2017). The peak learning rate is 10^{-3} with a linear warm-up of 500 batches, and the learning rate is decayed by half after 5 consecutive epochs of no improvement in validation loss. Weight decay 10^{-7} is used for additional regularization. Training a model takes approximately 12 hours on a single NVIDIA Quadro RTX 6000 GPU.

3.4 Auxiliary Phoneme Loss

One other improvement that we introduce to our EMG-to-speech model is an auxiliary phoneme prediction loss. For this loss, we predict a phoneme label for every time-step in addition to predicting the audio feature vectors at the output of the model. The relatively small data sizes available for this task creates a challenging learning problem, so the auxiliary loss is useful for providing additional guidance during training and regularizing the learned representations. Our use of an auxiliary phoneme loss for EMG-to-speech follows prior work that found phonemic prediction useful for related tasks like generating speech from ultrasound and ECoG sensors (Tóth et al., 2018; Anumanchipalli et al., 2019).

To get phoneme labels for each feature frame of the vocalized audio, we use the Montreal Forced Aligner (McAuliffe et al., 2017). The forced aligner uses

the reference text from our dataset along with a phonemic dictionary to determine likely phoneme sequences for each example. It then aligns the phoneme sequence with the example's audio by running a Viterbi decode through the sequence using an acoustic model for scoring. The accoustic model is a Gaussian mixture model associating phonemes with MFCCs that is pre-trained on the LibriSpeech dataset.

To predict a distribution over phonemes, we add an additional linear prediction layer and softmax on top of the transduction model encoder. For training, we modify the training loss by appending a term for phoneme negative log likelihood with a weight λ:

$$\mathcal{L} = \sum_i \left\| A'[i] - \tilde{A}'[i] \right\| - \lambda P[i]^\top \log \tilde{P}[i]$$

for audio feature targets $\tilde{A}'$, aligned audio feature predictions $\tilde{A}'$, one-hot phoneme target vector P, and aligned predicted phoneme probability vector $\tilde{P}$. We select $\lambda = .5$ for the phoneme loss weight by searching among values $\{5, 1, .5, .1, .05, .01\}$ and comparing validation performance. After training, the phoneme prediction layer is discarded.

We also found it useful to take phoneme predictions into account as part of the alignment when using the predicted-audio alignment. Just as the loss and the alignment costs $\delta[i, j]$ are identical for vanilla predicted-audio alignment, we can include the phoneme loss into δ when using our auxiliary loss to maintain the symmetry:

$$\delta_{audio+phoneme}[i, j] = \left\| A'_V[i] - \hat{A}'_S[j] \right\| - \lambda P_V[i]^\top \log \hat{P}_S[j]$$

3.5 Vocoding

The final component of our system is the vocoder model, which turns the audio features predicted by the rest of the model into raw audio waveforms that can be played through a speaker. The vocoder we use in our work is the HiFi-GAN vocoder (Kong et al., 2020), a neural model that is trained to predict audio waveforms on samples of voices. The HiFi-GAN generator model is a convolutional neural network model that generates all the audio samples in parallel, in contrast to autoregressive models like WaveNet (van den Oord et al., 2016) which generate one sample at a time.

As indicated by it's name, HiFi-GAN training is based primarily on a generative adversarial loss (Goodfellow et al., 2014), where a set of trained discriminators attempt to distinguish generated samples from real ones and the generator model is trained to try to fool the discriminators. HiFi-GAN uses a group of eight different discriminators that operate over different periods and scales to capture different views of the generated audio, making sure each of these views appear similar to real audio. HiFi-GAN also uses a combination of several other losses to improve the stability of training, including a discriminator feature matching loss and a spectrogram reconstruction loss.

To get high-quality outputs for the speaker in our data, we use a model pre-trained on many different speakers and then fine-tune on the vocalized examples from our own dataset. The pre-trained model was trained by the HiFi-GAN authors on a combination of the Librispeech (Panayotov et al., 2015), VCTK (Yamagishi et al., 2012), and LJSpeech (Ito and Johnson, 2017) datasets, which together contain over 1000 total hours of audio and thousands of different speakers. When fine-tuning, we use mel-spectrograms predicted by our transduction model as inputs rather than mel-spectrogram features from the true audio of the example. This use of predicted features lets the vocoder learn to overcome some artifacts in our predictions. Fine-tuning was run for 75 thousand steps using the default training hyperparameters and $V1$ model configuration from the HiFi-GAN paper.

We also explored several other vocoders such as WORLD (Morise et al., 2016) and WaveNet (van den Oord et al., 2016), but found that HiFi-GAN generated the most natural-sounding speech audio. In addition, running inference with HiFi-GAN is much faster than with WaveNet, since the autoregressive WaveNet model must generate one sample at a time.

Chapter 4

Evaluation

In this chapter, we will describe our experiments to measure the quality of audio outputs generated by our model from silent EMG signals. We will compare different variants of our model to understand which model configuration works best and compare our methods to prior work on EMG-to-speech synthesis.

The primary trait we aim to measure is intelligibiliy, or the number of words that can be understood from the generated audio. One way to measure intelligibility is to have humans write down what they hear in a sample and compare that to the reference text that was read during the creation of the dataset. We compare the evaluator's transcriptions and the reference with a word error rate metric (WER), which is computed as follows:

$$\text{WER} = \frac{\text{substitutions} + \text{insertions} + \text{deletions}}{\text{reference length}}$$

Lower WER values indicate better models. Prior to computing WER, text from the transcriptions and reference are normalized by removing punctuation and capitalization. We also perform a variant of this evaluation automatically by replacing the human transcription with the text predicted by an off-the-shelf automatic speech recognition system.

First, we will discuss our results using the automatic metric. Because this automatic metric can be run more quickly and consistently, we will use this evaluation to assess the various modeling decisions we have made. Then, we will look at results from a human evaluation. For the human evaluation, we will look at results in both an open-vocabulary domain from books and a closed-vocabulary domain of date and time expressions. Finally, we will talk about

a few other experiments exploring other questions such as the effect of data size, different electrode locations, cross-session generalization, and real-time streaming.

4.1 Automatic Evaluation

We will start our experiments with an automatic evaluation, which will allow us to easily compare a variety of model choices. The idea behind the automatic evaluation is to use an off-the-shelf automatic speech recognition (ASR) system as a proxy for humans listening to our samples. Although this evaluation will not perfectly capture the intelligibility to actual humans, improvements in the automatic metric do appear to be fairly correlated with human metrics, and we will validate our main results with human evaluators in Section 4.2 below. Our automatic evaluation is similar to the ASR evaluation used by Janke and Diener (2017), but with an off-the-shelf large-vocabulary ASR system instead of a limited-vocabulary system trained specifically on audio synthesized from EMG.

The ASR system we use for this evaluation is an open source implementation of DeepSpeech from Mozilla[1] (Hannun et al., 2014). The DeepSpeech system uses an acoustic model that predicts characters by training with the connectionist temporal classification (CTC) loss, and then combines this acoustic model with a language model during inference. The pre-trained models provided with this implementation were trained on several thousand hours of audio from a range of different domains. Running the recognizer on the original vocalized audio recordings from our test set results in a WER of 9.5%, which represents an approximate lower bound for this evaluation.

Test and validation examples used for evaluation were selected randomly from among the silent utterances, with 100 and 200 examples respectively in the open-vocabulary setting. The testing and validation examples use different utterances from those seen during training, and the vocalized examples with the same utterances are not used to avoid biasing the model.

One other thing to keep in mind when interpreting results is the variance in model performance across different training runs with different random initializations of model parameters. We found that results could vary by one or

[1] https://github.com/mozilla/DeepSpeech

two percentage points across different runs, so results should only be considered accurate to that degree. All of the important results here have differences much greater than the range attributable to this variation. For the most part, results are reported based on a single training run, but for the final test result of our full model we select the model with the best validation performance from among three runs with different initializations.

We will first compare different alignment costs for training on silent speech, then evaluate our other model improvements, and finally compare our best model to baselines representative of prior work. Our experiments in this section will focus on the open-vocabulary domain data, but we will look at closed-vocabulary results as part of the human evaluations below.

4.1.1 Comparing Alignment Methods for Silent Speech Training

In our first experiment, we aim to compare the different alignment methods described in Section 3.3, which allow us to train on silent speech by matching silent utterance frames with vocalized utterance frames. Each of our four methods use different information to form the alignment, resulting in different alignment costs δ. The four types of alignment we test are: EMG alignment (§ 3.3), EMG alignment with CCA (§ 3.3.1), audio alignment (§ 3.3.2), and audio-with-phoneme alignment (§ 3.4). The model we use for this experiment uses learned features (§ 3.1.2), a Transformer architecture (§ 3.2.2), and an auxiliary phoneme loss (§ 3.4).

Table 4.1 shows the validation error rates for each of the four alignment methods. We see that the audio-based alignment methods perform better than EMG-based alignment. Using the combination of audio feature and phoneme outputs for alignment performs the best, so this method will be used in all of the following experiments that train on silent utterances.

4.1.2 Evaluating Model Components

Our next experiment will evaluate three other major model components that we improved over prior work: the input features, model type, and auxiliary phoneme loss. Starting with our best model, which uses learned features, a Transformer model, and an auxiliary phoneme loss, we ablate each component one at a time to measure its effect on performance. We ablate the learned

Alignment Method	WER
EMG	58.5
EMG with CCA	48.7
Audio	40.0
Audio with phoneme	36.2

Table 4.1: Results with different alignment methods on the open-vocabulary validation set. Lower WER is better.

Model	WER
Full model	36.2
Ablation: Replace learned features with hand-designed features	44.4
Ablation: Replace Transformer with LSTM	44.4
Ablation: Remove auxiliary phoneme loss	46.0

Table 4.2: Ablations of model components. Results are open-vocabulary validation word error rates from an automatic intelligibility evaluation.

feature extraction by replacing the convolutional feature-extraction layers with hand-designed features, and we ablate the Transformer layers by replacing with LSTM layers. To ablate the phoneme loss, we simply set its weight in the overall loss to zero, including its weight in the alignment cost δ.

Table 4.2 shows the validation error rates for each of these ablations. All three ablations show an impact on our model's results, validating the usefulness of our model changes.

4.1.3 Comparing to Prior Work

Now that we have established which of our model variants perform best, let us compare our full model to methods used by prior work to see the full extent of our improvements.

As a baseline for this experiment, we use a model representative of the prior EMG-to-speech work by Janke and Diener (2017). Janke and Diener (2017) is the most recent prior EMG-speech-synthesis paper we are aware of with

the exception of Diener et al. (2018), whose methods are tailored specifically towards a different style of electrode setup with electrode arrays. Since prior work in this space reported numbers on private data and didn't release code, we can't directly compare to their implementations, so we will instead compare to our own implementation of a model that is broadly similar. At a high level the model in this past work follows a similar setup as ours, with a neural model that predicts time-aligned speech features from EMG features, followed by a vocoder to synthesize audio waveforms. In general, we try to err on the side of using stronger models with hyperparameters tuned on our own data rather than directly using the model sizes from past work. Because our dataset size is larger, the hyperparameters they used are often suboptimal in our setting.

The most important difference between prior work and ours is that the past work is trained only on EMG from vocalized speech. While one might hope that a model trained in this way could directly transfer to silent EMG, our results show that such an approach performs quite poorly due to differences between the two speaking modes. Most prior papers on EMG-to-speech only evaluated on vocalized data, but Toth et al. (2009) did attempt testing on silent speech and also observed a substantial degradation in quality after training only on vocalized data.

The model itself for our baseline uses the manual features described in Section 3.1.1, which come primarily from the same time-domain features used by Janke and Diener (2017). We also include features from a STFT in our baseline, which only improved performance in our experiments. For the neural network converting EMG features to speech features, we use an LSTM-based model in our baseline. Janke and Diener (2017) explored both LSTMs and feedfoward (multi-layer perceptron) networks in their experiments, and while they found the feed-forward network to work better on their data, we found LSTMs to work much better than feed-forward with our larger data size. We also use a much larger layer size for the LSTM, which we tuned on our validation set. One final difference between the baseline and our full model is that it does not include our auxiliary phoneme loss, since we are the first to use this type of loss for the EMG-to-speech task.

For our experiments, we use the same HiFi-GAN vocoder model for both the baseline and our full approach, though this is actually a much more recent vocoder than the MLSA vocoder used by Janke and Diener (2017) and other prior EMG-to-speech work. The largest difference in output between the HiFi-GAN vocoder and older vocoders like MLSA is in improved naturalness rather

Model	**WER**
Baseline model with vocalized training	88.3
Our full model	**36.1**
Silent training, without other improvements	67.8
Vocalized training, with other improvements	84.4

Table 4.3: Results comparing our full model with a baseline representative of prior work. Values come from the test set of the open-vocabulary data with an automatic intelligibility evaluation. Lower WER is better.

than intelligibility, but it may also improve intelligibility somewhat.

The test-set results comparing the baseline system with our full model are shown in Table 4.3. We see that this baseline trained on vocalized data does not perform well when tested on silent speech, with an error rate of 88.3%. For comparison, this same baseline model gets an error rate of 44.9% when tested on vocalized speech, showing the importance of considering differences between the two speaking modes. Overall, our methods improved error rates from 88.3% to 36.1%. This table also gives results with our silent training and other model improvements separately applied to the baseline to show the contribution of each component. Vocalized training continues to perform quite poorly even with our other model improvements, but the other model improvements do make a substantial difference when using silent training.

One other relevant result for comparison is that when both training and testing on vocalized speech, our best model reaches an error rate of 23.3% (using learned features, a Transformer, and an auxiliary phoneme loss). This result tells us that even with our silent speech training, we are still not able to reach the same level of accuracy on silent speech as vocalized speech. However, it is unclear whether this difference comes from the difficulty in training on silent speech, indicating there is more room for methodological improvements, or if there is less information available in silent speech inputs.

4.2 Human Evaluation

While the automatic evaluations we have run so far are useful for running experiments quickly and for ensuring consistent evaluation, they might not be an entirely accurate measure of intelligibility to humans. To validate our results further, we also run an intelligibility test with human evaluators. We perform this human evaluation on both the open-vocabulary and closed-vocabulary settings.

4.2.1 Open-Vocabulary Condition

In the open-vocabulary condition, we evaluate both our full model and the baseline model with vocalized training described in Section 4.1.3 above. Two human evaluators without prior knowledge of the text were each asked to listen to synthesized samples and write down the words they heard. We then compared these transcriptions to the ground-truth reference with a WER metric. The evaluators were each given 40 samples from our full model and 7 samples from the baseline. The text instructions given to the evaluators are as follows:

> Please listen to each of the attached sound files and write down what you hear. There are 40 files, each of which will contain a sentence in English. Write your transcriptions into a spreadsheet such as Excel or Google sheets so that the row numbers match the numbers in the file names. Many of the clips may be difficult to hear. If this is the case, write whatever words you are able to make out, even if it does not form a complete expression. If you are not entirely sure about a word but can make a strong guess, you may include it in your transcription, but only do so if you believe it is more likely than not to be the correct word. If you cannot make out any words, leave the corresponding row blank.

The results of this evaluation were a word error rate of 95.1% for the baseline model and 32.3% for our best approach. Thus, the evaluators were only able to make out a small number of words for the baseline but could hear much more in our model outputs.

One observation from this and other human evaluations we performed is that humans did worse than the automatic transcription on poor-performance

models but better than the automatic transcription for more intelligible models. On the baseline where humans scored 95.1% WER the automatic metric scored 88.3% WER, but on our best model where humans scored 32.3% WER the automatic transcription got 36.2% WER.

Another observation from this evaluation was a large variance in transcription error rates across different human evaluators. Even though the evaluators were listening to the exact same audio samples, the resulting error rates were very different for each evaluator: 36.1% and 28.5%. This large human variance suggests that the automatic metric may be more appropriate for establishing consistent evaluations, and we recommend that the automatic metric is used as the primary evaluation in comparisons by future work.

4.2.2 Closed-Vocabulary Condition

In the open-vocabulary results we've seen so far, the outputs still have a substantial number of words that cannot be understood. In this section, we'll look at how well our model can perform if we make the task easier by restricting the utterances to a closed-vocabulary domain of date and time expressions. Although we do not explicitly enforce any constraints on the model, by training on utterances from the closed-vocabulary data the model is able to learn to predict within this constrained set of outputs. The transduction model was trained on open-vocabulary data before being fine-tuned on the closed-vocabulary training set, and the vocoder model was trained only on open-vocabulary data. All other model architecture and training hyperparmeters were the same as our open-vocabulary model.

To asses intelligibility, a single human evaluator was given a set of 20 audio output files from our model and was asked to write out in words what they heard. The evaluator was told that the examples will contain dates and times, but was not given any further information about what types of expressions may occur. The text instructions given to the evaluator were as follows:

> Please listen to each of the attached sound files and write down what you hear as best you can. There are 20 files, each of which will contain an expression of some date or time. Write your transcriptions into a spreadsheet such as Excel or Google sheets so that the row numbers match the numbers in the file names. Although many of the clips will contain numbers, please write out what you

> hear as words. For example, you might write something like: `five oh two pm on Thursday`.[2] Some of the clips may be difficult to hear. If this is the case, write whatever words you are able to make out, even if it does not form a complete expression. For example: `five two pm on`. If you cannot make out any words, leave the corresponding row blank.

On model outputs from our best model, the word error rate of the human transcriptions was just 3.6%. This indicates that that vast majority of words can be understood in the audio synthesized for this closed-vocabulary condition.

4.3 Additional Experiments

In this section, we perform a few additional experiments exploring questions such as the effect of data size, different electrode locations, cross-session generalization, and real-time streaming. These experiments are all evaluated using the open-vocabulary automatic transcription method described in Section 4.1.

4.3.1 Data Size

In this section we explore the effect of dataset size on model performance. We train the EMG-to-speech transduction model on various-sized fractions of the dataset, from 10% to 100%, and plot the resulting WER. We select from the parallel (silent and vocalized) and non-parallel (vocalized only) portions proportionally here, but will re-visit the difference below. Although data size also affects the vocoder quality, we use a single vocoder trained on the full dataset for all evaluations to focus on EMG-specific data needs.

Figure 4.1 shows the resulting intelligibility measurements for each data size. As would be expected, there is a large improvement with small data sizes and the improvements slow down with more data. There continue to be some improvements for over 10 hours of data, though.

We also train a model without the non-parallel vocalized data. A model trained without this data has a WER of 44.4%, a loss of 8 absolute percentage

[2]We intentionally used an example that does not match a pattern in our generation procedure to avoid biasing the evaluator.

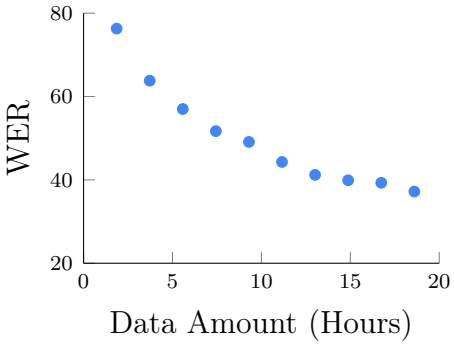

Figure 4.1: Effect of data amount on intelligibility.

points. This confirms that non-parallel vocalized data can be useful for silent speech even though it contains only data from the vocalized speaking mode. However, if we compare this result to a model where the same amount of data is removed proportionally from the two data types (parallel and non-parallel), we see that removing a mixture of both types leads to a much larger performance decrease to approximately 50% WER. This indicates that the non-parallel data is less important to the performance of our model, and suggests that future data collection efforts should focus on collecting parallel utterances of silent and vocalized speech rather than non-parallel utterances of vocalized speech.

4.3.2 Electrode Importance

In this section, we experiment with models that operate on a reduced set of electrodes to gain information about which electrodes are most important and assess their impact on performance. We perform a random search to try to find a subset of four electrodes that works well. More specifically, we sample 35 random combinations of four electrodes to use (out of 70 possible combinations) and train a model that only uses the selected subset. We then compare the models using their word error rate on the validation set. The motivation for using this form of analysis where multiple electrodes are dropped at a time is that electrodes may contain redundant information with each other and removing just a single electrode may be overly affected by this redundancy. Removing a single electrode often does not have enough effect on performance to distinguish from model initialization noise.

Electrodes Used	WER
1, 2, 3, 8	45.1
2, 3, 4, 7	46.0
1, 2, 3, 7	46.6
1, 2, 4, 5	47.1
2, 3, 6, 7	47.3
2, 3, 4, 6	47.6

	Location
1	left cheek just above mouth
2	left corner of chin
3	below chin back 3 cm
4	throat left of Adam's apple
5	mid-jaw right
6	right cheek just below mouth
7	right cheek 2 cm from nose
8	back of right cheek

Table 4.4: The best-performing size-four subsets of electrodes found in our random search. Word error rate is reported from the validation set.

Model performance with different four-electrode subsets ranged from around 45% to 54% WER – 9 to 18 points worse than the model with all electrodes. The best-performing models from this experiment are shown in Table 4.4. Many of the differences between these best results are within the range of difference attributable to random noise, so we shouldn't infer too much from the precise ranking. However, we can see some clear patterns of electrodes that appear in many of these top-ranked selections. If we order electrodes by the number of times they appear in a top-ranked selection, we get the following order: 2, 3, 1, 7, 4, 6, 8, 5. From this, we might infer that the two electrodes on and below the chin are the most important. These electrodes could be picking up tongue movement, which is one of most critical features of speech. The electrodes on the outside of the jaw and at the far back of the cheek seem to be least important, showing up just once on the list (though oddly the back cheek is included in the best model combination).

4.3.3 Cross-Session Results

For the test set used in the primary results above, examples are selected randomly from among the silent set of utterances. This means that these test examples will come from the same sessions with identical electrode placements as examples that were used for training. To assess the robustness of our model to small changes in electrode placement across sessions, here we run an evalua-

tion with a different test split where the test set comes from a separate session that is never trained on. We evaluate with four different test sessions, training a new model for each with all other sessions used for training. We use the entire held-out session as the test set, and randomly select 50 examples from among the other sessions as the validation set used for learning rate scheduling (where we decay the learning rate after a plateau in validation loss). The experiment is otherwise identical to the full model results above using the automatic evaluation. The resulting error rates on the new sessions are 41.3%, 41.8%, 43.6%, and 47.3%, ranging from approximately five to eleven points worse than results with in-session training. This drop in accuracy shows the importance of considering methods to improve robustness or adapt quickly to session differences, a direction which some other work has begun to explore on the related EMG-to-text task (Maier-Hein et al., 2005; Wand et al., 2018; Proroković et al., 2019).

4.3.4 Real-Time Streaming

Many use cases for voicing of silent speech would be best served by models that operate in a real-time streaming mode, where audio is generated for each sound immediately after the corresponding mouth movement is detected. However, the models we have seen so far have operated in an offline mode, where a complete utterance is processed after the user has finished speaking. These models aren't quite compatible with a real-time streaming mode because model decisions in the middle of the utterance have access to information from inputs later in the sequence, which we will not have seen yet when streaming.

To adapt our models to work in a real-time streaming mode, several changes must be made so that the models only rely on information prior to the time-step being processed. First, we must adapt the signal filtering from a forward-backward filter to a left-to-right filter. Then, we modify the convolution layers in both the transduction and vocoder models to use causal convolutions, which are shifted to only look at a window up to the current time-step rather than both directions. Finally, we mask the Transformer attention in a similar way, so that only the current and past time-steps can be attended to. After making these changes, our model's intelligibility was reduced by approximately twelve percentage points to a WER of 47.8%.

We also made an initial prototype where this streaming model is applied in real-time as signals are captured. The model was able to generate speech

when used in this way, showing that this mode of operation is viable. However, the latency of our prototype was quite large and it took over a second for sounds to be produced after being mouthed. Further work would be needed to optimize the latency, which could require delay reduction in each step of the processing pipeline, including the streaming of signals from the data-collection board, the model processing, and the audio playback. A total mouth-to-ear latency of 150 to 200 ms is considered acceptable for phone calls (International Telecommunication Union, 2003), so to be useful in a phone application the system delay plus telecommunication delay would need to be below that range. To be played back to the speaker a delay below 50 ms would be more acceptable, since delays over that amount are known to cause changes in speaking behavior (Stuart et al., 2002).

Chapter 5

Phoneme Error Analysis

In this chapter, we perform an analysis of what our models have learned. To do this analysis, we will look at the outputs of our auxiliary phoneme prediction task, since these outputs provide an interpretable view of what the model knows. Although the phoneme predictions are not directly part of the audio synthesis process, we have observed that mistakes in audio and phoneme prediction are often correlated. To analyze the phoneme predictions for silent utterances, we align each frame of the parallel vocalized utterance with a silent frame using the predicted audio and phonemes, as described in Sections 3.3.2 and 3.4. Then, we compare phonemes for paired frames to perform two types of analysis: understanding which phoneme pairs get confused and evaluating accuracies along particular articulatory feature dimensions.

5.1 Confusion

For our first analysis, we look at the confusion between each pair of phonemes, or how often one phoneme is predicted in the place of another. We use a symmetric frequency-normalized metric for confusion.

$$\text{Confusion: } (e_{p1,p2} + e_{p2,p1})/(f_{p1} + f_{p2})$$

where $e_{p1,p2}$ is the number of times $p2$ was predicted when the aligned label was $p1$, and f_{p1} is the number of times phoneme $p1$ appears as a target label. Figure 5.1 illustrates this measure of confusion for each pair of phonemes using the darkness of lines drawn between them, and Table 5.1 lists the values of

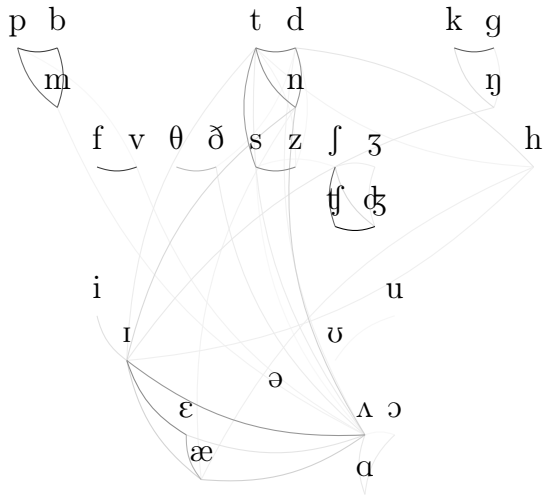

Figure 5.1: Phoneme confusability. Darker lines indicate more confusion, and the maximum darkness represents 13% confusion.

the most confused pairs. The table also includes a measure of the accuracy for each pair to give an idea of how often those phonemes are mistaken in general rather than just with each other.

$$\text{Accuracy: } (e_{p1,p1} + e_{p2,p2})/(f_{p1} + f_{p2})$$

We observe that many of the confusions are between pairs of consonants that differ only in voicing. This is consistent with observations we have made on the signals themselves where the throat electrode is much more active during vocalized speaking than silent speaking, suggesting that voicing signals are subdued or missing in silent speech.[1] Another finding is that nasals and stops are often confused. This distinction is challenging due to the role of the velum (soft palate) and its relatively large distance from the surface electrodes, as has been noted in prior work (Freitas et al., 2014). Other confusions are also shown in the figure but are generally a bit harder to interpret.

[1]Given that Edfeldt (1959) found some vocal cord activation during subvocal speech when measuring with needle electrodes inserted into the larnyx, it seems likely there is still some activation during silent speech, but it may be too small to be easily picked up by our surface electrodes.

Phonemes		Confusion (%)	Accuracy (%)
ʤ	ʧ	13.2	49.4
v	f	10.4	72.0
p	b	10.3	64.3
m	b	9.3	74.3
k	g	8.9	77.2
ʃ	ʧ	8.3	59.8
p	m	8.1	73.0
t	d	7.2	64.0
z	s	6.6	80.0
ɪ	ɛ	6.5	60.6
t	n	6.3	67.1
n	d	6.0	66.8
ɪ	ʌ	6.0	65.8
ɹ	ɚ	5.7	78.2
t	s	5.5	72.8

Table 5.1: Numerical values for confusion and accuracy of the most commonly confused phoneme pairs.

5.2 Articulatory Feature Accuracy

To better understand how well different consonant articulatory features are captured by our model, we perform a second type of analysis where we ask the model to choose between sets of phonemes that differ along a particular feature dimension. For this analysis, we define a confusion set for an articulatory feature as a set of English phonemes that are identical across all other features. For example, one of the confusion sets for the place feature is {p, t, k}, since these phonemes differ in place of articulation but are the same along other axes like manner and voicing. For each feature of interest, we calculate a forced-choice accuracy within the confusion sets for that feature. More specifically, we first run our model on the silent input and align to a vocalized target, then find all time-steps in the target sequence with labels belonging in a confusion set for the feature being analyzed. For each predicted distribution aligned to one

Feature	Confusion Sets
Place	{p,t,k} {b,d,g} {m,n,ŋ} {f,θ,s,ʃ,h} {v,ð,z,ʒ}
Oral manner	{t,s} {d,z,l,r} {ʃ,ʧ} {ʒ,ʤ}
Nasality	{b,m} {d,n} {g,ŋ}
Voicing	{p,b} {t,d} {k,g} {f,v} {θ,ð} {s,z} {ʃ,ʒ} {ʧ,ʤ}

Figure 5.2: Phoneme confusion sets used for our articulatory feature analysis. The phonemes in each confusion set share the same articulatory feature values for all features except the one being tested.

of these time-steps, we select the highest-scoring phoneme label from within the corresponding confusion set. Using these new labels, we then compute an accuracy across all those positions that have a confusion set. Table 5.2 lists all confusion sets used for this analysis.

As a point of comparison for this analysis, we also run a baseline to try to determine how much of the feature accuracies can be attributed to information from phonemic context rather than information extracted from the EMG signals. This baseline measures a similar forced-choice accuracy across features, but using a model that is trained to make decisions based on nearby phonemes to try to capture phonotactic and language modeling constraints. In the place of EMG feature inputs, the baseline model is given a sequence of phonemes, but with information about the specific feature being tested removed by collapsing phonemes in each of its confusion sets to a single symbol. Figure 5.3 illustrates the inputs and output choices for this baseline on an example. The input phonemes come from the predictions of the full EMG-based model on silent examples, but after alignment to the parallel vocalized target sequence. We use predicted phonemes instead of gold phonemes as inputs because this represents information the model already knows about nearby phonemes. Because the input phonemes are pre-aligned to the vocalized targets, the baseline model and its loss do not need any additional alignment during training. The model itself uses a Transformer architecture with dimensions identical to our primary EMG-based model, but is fed phoneme embeddings of dimension 768

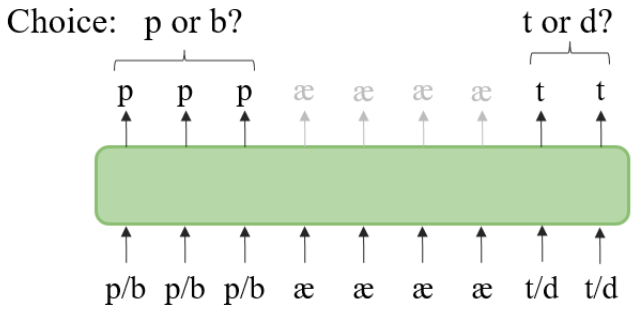

Figure 5.3: An example of the input tokens and output choices for the feature analysis phoneme-context baseline. In this example the feature of voicing is being tested, so phonemes like p and b that differ only by voicing are represented with the same token at the input, and the output is asked to choose between them. In this example it would be hard for the baseline model to predict which phoneme is correct given just the phonemic context, since *pat*, *bat*, *pad*, and *bad* are all valid words. If the full model with EMG input is able to distinguish such cases better than the baseline, that indicates it may be getting useful information about the feature from the EMG rather than just context.

in the place of the EMG features output from the convolutional layers. The model is trained with a cross-entropy loss over the full set of phonemes, but choices are restricted during evaluation in the same way as for the EMG-model analysis. We train a separate baseline model for each of the four articulatory feature types to account for different collapsed sets in the input. Other training hyperparameters are the same between this baseline and the main model.

The results of this articulatory feature analysis are shown in Figure 5.4. In addition to the full EMG-based model and the phoneme-context baseline, we also include a majority-class baseline for comparison, which simply uses the most common phoneme from each confusion set as the prediction for that set. By comparing the gap in accuracy between the full model and the phoneme-context baseline, we observe similar trends to those we saw in our confusion analysis. While place and oral manner features can be predicted much better by our EMG model than from phonemic context alone, nasality and voicing are more challenging and have a smaller improvement over the contextual baseline.

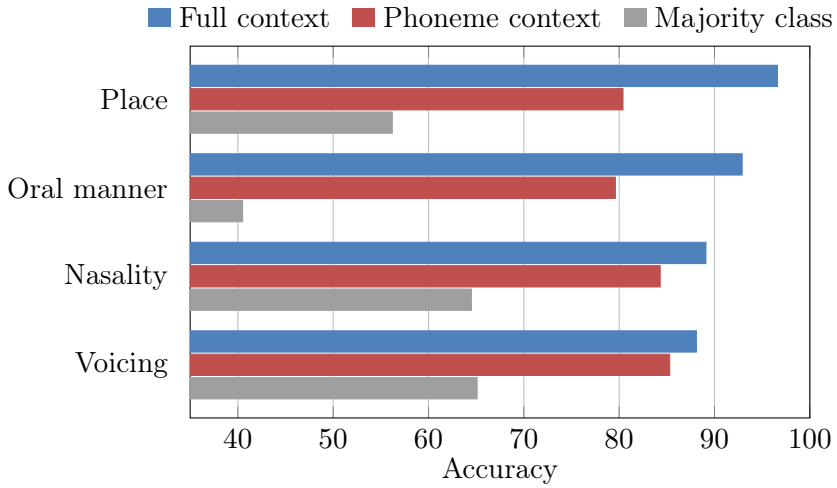

Figure 5.4: Accuracy of selecting phonemes along articulatory feature dimensions. We compare our full EMG model (full context) with a majority-class baseline and a model given only phoneme context as input.

More analysis is going to be needed to figure out exactly how much we can pick up from the larynx and soft palate, but we do see some improvement over the contextual baseline, which may indicate we are picking up at least some useful information for these features from EMG.

Chapter 6

EMG Speech Recognition

So far, this thesis has focused on the problem of voicing, or directly synthesizing audio from silent EMG. In this chapter, we will take a brief look at the related problem of EMG speech recognition, where our output will instead be text of what was said. This speech recognition task is useful for a slightly different set of applications like talking to a computer or phone, whereas voicing is more useful when talking to other people, as discussed in Section 1.2.

While our evaluation from Section 4.1 did end up extracting text from our outputs, that extraction was intended as a proxy for human listeners when evaluating the audio outputs, and our end goal was to improve intelligibility to humans. In this chapter, we will explore how we might get text more directly when text is the output of interest. This direct text prediction will be more efficient and also slightly more accurate than getting text indirectly through audio synthesis and audio-based ASR.

While there has been a substantial body of prior work on EMG speech recognition in the past, that work used older HMM-based recognition techniques and operated over limited vocabulary sizes (Wand and Schultz, 2011; Meltzner et al., 2018) (see Section 1.4). Our work instead uses state-of-the art neural methods like Transformers and a CTC loss, and operates over a large open vocabulary.

6.1 Methods

Our model for speech recognition will be a character-prediction model trained with a connectionist temporal classification (CTC) loss (Graves et al., 2006). The model predicts a character output for each frame of the input, then collapses the output sequence according to the rules defined by CTC, as described below. The majority of the neural architecture for the model will be the same as for our transduction model in Section 3.2, with convolutional feature extractors followed by a sequence of Transformer layers. The only difference will be at the final model layer, which will now be a softmax over the character vocabulary instead of a linear projection to speech features. We normalize all text by lowercasing and removing punctuation, so the vocabulary will contain the 26 lowercase English letters, 10 digits, and a space character.

CTC collapses from the frame-level outputs of the model to the final output sequence by collapsing any contiguous sequences of repeated characters. In addition, it also introduces a special *blank* character representing no output, which is included as an option for the softmax of the model at each frame. To output two of the same character in a row, the model must include a blank character between them in the frame-level output.

The training loss for CTC is the sum of the probability of all frame-level output sequences that collapse to the correct label sequence, which can be computed efficiently with a dynamic program. During inference, beam search is used to search for the output sequence with the highest probability when summed over possible paths. A language model is also integrated during inference by multiplying language model probabilities with the character probabilities output from the model. We use the same language model as the DeepSpeech ASR system we used in our automatic evaluation (§ 4.1), which is a 5-gram language model with modified Kneser-Ney smoothing (Chen and Goodman, 1999) trained on the combination of several popular ASR datasets.

The model size hyperparameters are kept the same as for the voicing task, but we modify the training hyperparameters somewhat based on validation set tuning. We run training for 200 epochs and do not use weight decay regularization. The learning rate is warmed-up linearly over the first 1000 steps to 3e-4, then decreased by half at epochs 125, 150, and 175. We do not use the auxiliary phoneme prediction task during training.

6.2 Results

To evaluate the model, we use a WER comparison of outputs with the reference text from our dataset. It is generally the same as our evaluation from Section 4.1, except that now the text output comes directly from our model rather than from an external ASR system. On this evaluation, our model achieves a validation WER of 28.8%.

To put this result into context, we can compare to our best result from synthesizing audio and then using an external ASR system as in Section 4.1, which achieved a WER of 36.2% on the same examples. This comparison indicates it may be better for accuracy to directly predict text when that is the desired output, though there are several factors that could be affecting the comparison. One factor is that the DeepSpeech audio-based ASR system used in our cascaded system uses a slightly older LSTM-based architecture than our Transformer-based model. In addition, the cascaded model performance could likely be improved by fine-tuning on the outputs of our system, letting it adapt to the types of errors made by our synthesis model.

To get a better idea of the quality of EMG compared to audio as an input for speech recognition, we also train a recognition model on our data with audio as input. For this test, we replace the EMG feature inputs to our recognition model with mel-spectrogram audio features, but keep the same Transformer architecture for the body of the model. We train the model on the vocalized data from our dataset, which has both spoken audio and associated text. Although the 15.1 hours of data used for this training doesn't exactly match the 18.7 hours for the EMG-model training, it is roughly comparable. The audio-based model reaches a validation WER of 11.3%, compared to 28.8% with EMG inputs. Thus, it is still more challenging to decode speech from EMG than from audio.

Chapter 7

Conclusion

This thesis investigated methods for the task of voicing silent speech, where silently mouthed words are converted to audible speech based on electromyographic signals. This task has many potential real-world applications, from private phone conversations to restoring speech in some clinical settings.

Our work has made several contributions toward generating intelligible speech from silent EMG signals. First, we collected and publicly released a dataset for the task, which we hope will provide a consistent benchmark for future development. Next, we introduced a method for training on EMG signals from silent speech, allowing our model to better adapt to the particular features of that speaking mode. We also improved several other aspects of the model, such as better feature extraction, an improved model architecture, and additional learning signal. Finally, we performed experiments and analysis on our model to understand what it has learned and give more insights into its behaviors.

There are still many important areas for future work on the task of voicing silent speech. One important area is to extend to multi-speaker models, rather than focusing on a single speaker as we did in our work. For most real-world use cases, the model must work with little to no data for a new speaker and will likely need to generalize from a large dataset containing many different speakers. One other related area of work is to improve generalization across different sessions. Since each use of a model will occur with new electrode placements, it will be important to have methods for increasing robustness and adapting quickly to new locations. Another area for improvement is in the choice of where electrodes are placed. Choosing good locations is important for

optimizing signal quality and for integrating the electrodes into a streamlined device that can easily be put on and taken off. One last area for future work is in improving the input sensors and signal quality. Improving the input signals has the potential to greatly improve results on the task, and exploring different sensors or combinations of sensors could result in better ways of capturing silent speech. These are just a few of the many potential future directions for improving this technology and making it viable for real-world use.

Bibliography

Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. 2019. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498.

Adrian DC Chan, Kevin Englehart, Bernard Hudgins, and Dennis Fenton Lovely. 2002. Hidden markov model classification of myoelectric signals in speech. *IEEE Engineering in Medicine and Biology Magazine*, 21(5):143–146.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3444–3453. IEEE.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

L. Diener, G. Felsch, M. Angrick, and T. Schultz. 2018. Session-independent array-based EMG-to-speech conversion using convolutional neural networks. In *Speech Communication; 13th ITG-Symposium*, pages 1–5.

Lorenz Diener, Matthias Janke, and Tanja Schultz. 2015. Direct conversion from facial myoelectric signals to speech using deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Michael D'Zmura, Siyi Deng, Tom Lappas, Samuel Thorpe, and Ramesh Srinivasan. 2009. Toward EEG sensing of imagined speech. In *International Conference on Human-Computer Interaction*, pages 40–48. Springer.

Åke W Edfeldt. 1959. *Silent speech and silent reading.* Ph.D. thesis, Almqvist & Wiksell.

João Freitas, António JS Teixeira, Samuel S Silva, Catarina Oliveira, and Miguel Sales Dias. 2014. Velum movement detection based on surface electromyography for speech interface. In *BIOSIGNALS*, pages 13–20.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips. *Speech Communication*, 52(4):288–300.

International Telecommunication Union. 2003. ITU-T recommendation G.114.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Keith Ito and Linda Johnson. 2017. The LJ speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Muhammad Zahak Jamal. 2012. Signal acquisition using surface EMG and circuit design considerations for robotic prosthesis. In Ganesh R. Naik, editor, *Computational Intelligence in Electromyography Analysis*, chapter 18. IntechOpen, Rijeka.

Matthias Janke and Lorenz Diener. 2017. EMG-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2375–2385.

Chuck Jorgensen, D Lee, and Shane Agabon. 2003. Sub auditory speech recognition based on EMG/EPG signals. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1098–7576.

Szu-Chen Stan Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alexander H. Waibel. 2006. Towards continuous speech recognition using surface electromyography. In *INTERSPEECH*.

Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. AlterEgo: A personalized wearable silent speech interface. In *23rd International Conference on Intelligent User Interfaces*, pages 43–53.

Arnav Kapur, Utkarsh Sarawgi, Eric Wadkins, Matthew Wu, Nora Hollenstein, and Pattie Maes. 2020. Non-invasive silent speech recognition in multiple sclerosis with dysphonia. In *Machine Learning for Health Workshop*, pages 25–38. PMLR.

Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: an ultrasound imaging-based silent speech interaction using deep neural networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Peter Ladefoged and Keith Johnson. 2014. *A course in phonetics*. Cengage learning.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713.

Boon Pang Lim. 2011. *Computational differences between whispered and non-whispered speech*. University of Illinois at Urbana-Champaign.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lena Maier-Hein, Florian Metze, Tanja Schultz, and Alex Waibel. 2005. Session independent non-audible speech recognition using surface electromyography. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 331–336. IEEE.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2017. Silent speech recognition as an alternative communication device for persons with laryngectomy. *IEEE/ACM transactions on audio, speech, and language processing*, 25(12):2386–2398.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering*, 15(4):046031.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. 2016. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884.

Michael S Morse and Edward M O'Brien. 1986. Research summary of a scheme to ascertain the availability of speech information in the myoelectric signals of neck and head muscles using surface electrodes. *Computers in biology and medicine*, 16(6):399–410.

Y. Nakajima, H. Kashioka, K. Shikano, and N. Campbell. 2003. Non-audible murmur recognition input interface using stethoscopic microphone attached to the skin. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 5, pages V–708.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *ArXiv*, abs/1609.03499.

OpenBCI. 2014. Cyton board documentation. `https://docs.openbci.com/Cyton/CytonLanding/`.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Stavros Petridis and Maja Pantic. 2016. Deep complementary bottleneck features for visual speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2304–2308.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. 2018. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *International Conference on Learning Representations*.

Cathy J Price. 2012. A review and synthesis of the first 20 years of PET and fMRI studies of heard speech, spoken language and reading. *Neuroimage*, 62(2):816–847.

Krsto Proroković, Michael Wand, Tanja Schultz, and Jürgen Schmidhuber. 2019. Adaptation of an EMG-based speech recognizer via meta-learning. In

2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 1–5. IEEE.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of speech recognition*. Prentice Hall.

Ignacio Rodriguez-Carreno, Luis Gila-Useros, and Armando Malanda-Trigueros. 2012. Motor unit action potential duration: Measurement and significance. In Ihsan M. Ajeena, editor, *Advances in Clinical Neurophysiology*, chapter 7. IntechOpen, Rijeka.

Valerie C Scanlon and Tina Sanders. 2018. *Essentials of anatomy and physiology*. FA Davis.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *INTERSPEECH*.

Tanja Schultz and Michael Wand. 2010. Modeling coarticulation in EMG-based continuous speech recognition. *Speech Communication*, 52(4):341–353.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. 2022. Learning audio-visual speech representation by masked multimodal cluster prediction. *arXiv preprint arXiv:2201.02184*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Andrew Stuart, Joseph Kalinowski, Michael P Rastatter, and Kerry Lynch. 2002. Effect of delayed auditory feedback on normal speakers at two speech rates. *The Journal of the Acoustical Society of America*, 111(5):2237–2241.

Noboru Sugie and Koichi Tsunoda. 1985. A speech prosthesis employing a speech synthesizer-vowel discrimination from perioral muscle activities and vowel production. *IEEE Transactions on Biomedical Engineering*, BME-32(7):485–490.

Texas Instruments. 2012. ADS1299 datasheet. `https://www.ti.com/lit/ds/symlink/ads1299.pdf`.

Arthur R. Toth, Michael Wand, and Tanja Schultz. 2009. Synthesizing speech from electromyography using voice transformation techniques. In *INTERSPEECH*.

László Tóth, Gábor Gosztolya, Tamás Grósz, Alexandra Markó, and Tamás Gábor Csapó. 2018. Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces. In *INTERSPEECH*, pages 3172–3176.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Michael Wand, Matthias Janke, and Tanja Schultz. 2014a. The EMG-UKA corpus for electromyographic speech processing. In *INTERSPEECH*.

Michael Wand, Matthias Janke, and Tanja Schultz. 2014b. Tackling speaking mode varieties in EMG-based speech recognition. *IEEE transactions on biomedical engineering*, 61(10):2515–2526.

Michael Wand and Tanja Schultz. 2009. Towards speaker-adaptive speech recognition based on surface electromyography. In *Biosignals*, pages 155–162.

Michael Wand and Tanja Schultz. 2011. Session-independent EMG-based speech recognition. In *Biosignals*, pages 295–300.

Michael Wand, Tanja Schultz, and Jürgen Schmidhuber. 2018. Domain-adversarial training for session independent EMG-based speech recognition. In *Interspeech*, pages 3167–3171.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech 2017*, pages 4006–4010.

Alan Wrench and Korin Richmond. 2000. Continuous speech recognition using articulatory data. In *International Conference on Spoken Language Processing*. International Speech Communication Association.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. 2012. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit.

Feng Zhou and Fernando Torre. 2009. Canonical time warping for alignment of human behavior. In *Advances in neural information processing systems*, pages 2286–2294.