



VCU

Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations


Graduate School

2022

Universal Design in BCI: Deep Learning Approaches for Adaptive Speech Brain-Computer Interfaces

Srdjan Lesaja

Follow this and additional works at: <https://scholarscompass.vcu.edu/etd>

 Part of the [Artificial Intelligence and Robotics Commons](#), [Bioelectrical and Neuroengineering Commons](#), and the [Data Science Commons](#)

© The Author

Downloaded from

<https://scholarscompass.vcu.edu/etd/7154>

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

©Srdjan Lesaja, 2022
All Rights Reserved.

VIRGINIA COMMONWEALTH UNIVERSITY

DOCTORAL DISSERTATION

**Universal Design in BCI: Deep Learning
Approaches for Adaptive Speech
Brain-Computer Interfaces**

Author:

Srdjan LESAJA

Supervisor:

Prof. Dean J. KRUSIENSKI

*A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Department of Biomedical Engineering
College of Engineering

November 30, 2022

Acknowledgements

Regardless of how proud I am of the rest of this work, this is the section that is the most important to me. It's been rewritten many times with no version good enough for you, but now I've run out of time so this is the version that it must be.

I want to thank my parents for all that they've sacrificed to give me such opportunity. I want to thank my dad for giving me math, and with it instilling in me a love for abstract thought. I want to thank my mom for grounding me in the earth, showing me the beauty in all of the systems that surround us, and reminding me always to walk small beneath the big night sky. I want to thank my sister, for making me a battler, and for showing me what it means to never give up.

For this dissertation, I want to thank my committee. And my advisor, Dean; for all the lessons in and out of the lab. Thank you Chip and Mouse for all your help editing this document. I want to express my gratitude to my collaborators, colleagues, and fellow students. To Josh, who was there at the very start. To Xavier, for helping sharpen the why. To Miguel and Garrett, for the conversations. To Christian, for the help and mentorship. But especially, I want to thank Morgan, without whom this work would be a pale version of itself. In the self-determined labeling of clusters for my particular neural net, you stand apart, a class onto yourself.

I want to thank Laurie, for creating the circumstance that allowed me to ask myself the important questions. I will never forget your selfless benevolence. To all the helpers, who will never know the meaning and impact of their kindness, thank you for your gestures big and small.

I want to thank all of my friends: the seven, Elsa and 1705, D&D, D&B, B&B, Chimbo, the AA's, Erich, Steven; too many to list. I have grown into the person that I am with you. Each of you has made a difference in my life. You are all such wonderful humans, and it's a privilege to know you. I will always cherish your company, and all the times we've shared together.

I want to thank Michelle and Anna, for your unyielding support, care, encouragement, and for being a bright light to ferry me through dark times. You've kept me sane, and given me perspective, acceptance, and the will to persevere. Words could never do it justice; I will have to find another way to convey the gravity of what all you've done means to me.

Last but not least, I want to thank Gingie, for being literally beside me day in day out, even as I write this. I will spend a lifetime repaying you for all the sunshine time you forwent these last years, guarding ever-vigilantly as I clacked away at the black box.

I feel so lucky and grateful that you are all in my life.

The problem is, we think we have time...

VIRGINIA COMMONWEALTH UNIVERSITY

Abstract

Department of Biomedical Engineering
College of Engineering

Doctor of Philosophy

Universal Design in BCI: Deep Learning Approaches for Adaptive Speech Brain-Computer Interfaces

by Srdjan LESAJA

In the last two decades, there have been many breakthrough advancements in non-invasive and invasive brain-computer interface (BCI) systems. However, the majority of BCI model designs still follow a paradigm whereby neural signals are preprocessed and task-related features extracted using static, and generally customized, data-independent designs. Such BCI designs commonly optimize narrow task performance over generalizability, adaptability, and robustness, which is not well suited to meeting individual user needs.

If one day BCIs are to be capable of decoding our higher-order cognitive commands and conceptual maps, their designs will need to be adaptive architectures that will evolve and grow in concert with users, as well as the ever-progressing landscape of technological innovation.

Speech is a complex neural process, involving planning, motor execution, auditory self-perception, and semantic encoding. This makes speech an attractive target for the development of adaptive BCI. Non-invasive BCIs, such as those utilizing scalp EEG, lack the spatial resolution and spectral bandwidth required for decoding of complex dynamics of speech processes. The present work uses intracranial signals, from stereotactic EEG and electrocorticography, which possess signal characteristics better suited for the development of practical speech BCIs.

Deep learning is a machine learning approach in which features and the classifier can be jointly learned directly from the data. Such approaches have been demonstrated to be uniquely able to model applications involving high-dimensional, unstructured data, such as computer vision and natural language processing.

This work argues for universal design principles and deep learning architectures as the foundations for the development of robust, user-centered BCI. First, it is shown that combining traditional feature extraction techniques with deep learning models does not confer performance benefits, and comes at the cost of computational inefficiency and increased barriers to reproducibility. Then, a

novel model is presented, SincIEEG, for speech activity detection from intracranial neural signals. Initial model layers learn data-driven features corresponding to frequency bands. The interpretable features are used to show that models derive person-specific features. Additionally, results confirm that conventional feature extraction methods are excluding frequency bands useful for detecting speech. Furthering analysis of SincIEEG, the transfer learning potential of the model is systematically quantified, and hyperparameters that have the greatest impact are summarized.

Finally, the power of deep learning and data-driven modeling is showcased. We present a first-of-its-kind modeling framework in HUBRIS; a self-supervised, transformer-based, transfer learning approach, capable of training from unlabeled data pooled from multiple participants. This is enabled partly by a novel embedding of the neuroanatomical electrode locations in the model. Models learn self-derived pseudo-lexical speech representations and are evaluated using three disparate downstream speech classification tasks to highlight the generalizability of this design.

Contents

Acknowledgements	iii
Abstract	vii
1 Introduction	1
1.1 Organization	4
1.1.1 Contributions	5
2 Background	11
2.1 The Brain	12
2.1.1 Speech processes in the Brain	15
2.2 Brain-Computer Interfaces	18
2.2.1 Properties of BCI Sensing Modalities	18
2.2.2 Stereotactic EEG and Electrocorticography	22
2.2.3 Speech Processes from Invasive BCIs	26
2.3 Deep Learning	28
2.3.1 Supervised Learning	31
2.3.2 Deep Learning Model Design and Training	32
2.3.3 Deep Learning for Speech BCI	36
3 Intracranial EEG Datasets	39
3.1 Single Word Experiment - Electrocorticography	40
3.2 Harvard Sentences Experiment - Stereotactic EEG	44
4 Comparison of Feature Extraction Methods for Speech BCI	49
4.1 Introduction	49
4.2 Background	51
4.2.1 Convolutional Neural Networks (CNNs)	52
4.2.2 Prior Work	53
4.3 Methods and Model	54
4.3.1 Dataset	54
4.3.2 Signal Processing Protocols	56
4.3.3 Method of Comparison	59
4.3.4 CNN Models	59
4.4 Results	62

4.5	Discussion	65
5	SincIEEG: Learning Person-specific Features	67
5.1	Introduction	67
5.2	Materials and Methods	69
5.2.1	Dataset	69
5.3	Model Design and Optimization	70
5.3.1	Multi-SincNet Input Convolution	71
5.3.2	Activation	73
5.3.3	Batch Normalization	73
5.3.4	Monte Carlo Dropout	74
5.3.5	Optimization Procedure	75
5.4	Model Validation	76
5.4.1	Prediction Accuracy	76
5.4.2	Spectral Band Convergence	77
5.4.3	Comparison Models and Benchmarks	77
5.5	Results	79
5.5.1	Prediction Accuracy	79
5.5.2	Spectral Band Convergence	82
5.5.3	Comparison and Benchmarks	85
5.6	Discussion	86
6	Transfer Learning for Speech Activity Detection	89
6.1	Introduction	89
6.2	Background	91
6.2.1	Transfer Learning	91
6.3	Methods	94
6.3.1	Datasets	94
6.3.2	Model	96
6.3.3	Transfer Learning Protocol	97
6.3.4	Transfer Learning Experiments	98
6.4	Results	102
6.4.1	Between Participant	102
6.4.2	Between Task	107
6.5	Discussion	111
7	Hidden Unit Brain Representations from Intracranial Signals	115
7.1	Introduction	116
7.2	Background	118
7.2.1	Self-Supervised Learning	118
7.2.2	Transformers	119
7.3	sEEG Dataset - Harvard Sentences	121
7.3.1	Volumetric Morphing of Electrode Locations to a Common Brain Atlas	122

7.4	Self-supervised pretraining methodology	123
7.4.1	Model Architecture	124
7.4.2	Pretraining	130
7.5	Evaluation on Classification Tasks	132
7.5.1	Leave-one-participant-out Pretraining	132
7.5.2	Downstream Classification	134
7.6	Results	136
7.7	Discussion	140
8	Conclusions and Future Work	147
9	Curriculum Vitae	153
	Bibliography	155

List of Figures

2.1	Diagram of brain structures	14
2.2	Diagram of speech neural circuit	17
2.3	Signal properties of BCIs	19
2.4	Neuron dipole moment and local field potentials	21
2.5	ECoG and sEEG Procedures	24
2.6	Biological and artificial neurons.	29
2.7	Common deep learning architectures	35
2.8	Common activation functions	35
3.1	Diagram of Single Word experiment	40
3.2	Electrode locations for all 5 participants. Electrodes identified in the auditory cortex region are highlighted in red.	42
3.3	Harvard Sentences experiment protocol diagram	44
3.4	Electrode locations for the Harvard Sentences experiment.	47
4.1	Diagram of traditional BCI modeling framework.	51
4.2	Diagram of CNN convolutional layer	52
4.3	Speech activity detection labeling scheme for Single Word data	55
4.4	Diagram of the Herff signal processing method.	57
4.5	Diagram of the Moses signal processing method	57
4.6	Diagram of the three signal processing methods	60
4.7	Base CNN model architecture	61
4.8	Stratified performance of models for comparison of signal processing methods.	63
5.1	The SincIEEG model architecture	70
5.2	SincIEEG performance across model configurations	80
5.3	SincIEEG model prediction examples	81
5.4	Spectral band convergence plot	82
5.5	SincIEEG learned frequency bands by participant	83
5.6	SincIEEG learned frequency band across all participants	84
6.1	Transfer learning diagram	92
6.2	Harvard Sentence imagined speech labeling scheme	96
6.3	Box plot for Between-participant transfer learning	105

6.4	Pretrained model accuracies effect on fine-tune accuracy for the Between-participant experiment	106
6.5	Pretrained model accuracies effect on fine-tune accuracy for the Between-participant experiment	107
6.6	Model Accuracies between Imagining and Speaking tasks	109
6.7	Box plot of residual accuracy for the Between-Task experiment	110
6.8	Box plot of residual accuracy for the Between-Task experiment	111
7.1	Diagram of Transformer layer	120
7.2	Electrode location morphed to common brain atlas	122
7.3	HUBRIS pretraining architecture	125
7.4	Example of masking procedure	129
7.5	Diagram of the downstream task training procedure	133
7.6	Box plot of accuracy across participants for the 3 downstream tasks	137
7.7	Cross-validation loss of HUBRIS model over pretraining epochs.	139
7.8	Confusion matrices of fine-tuning classification tasks	140
7.9	t-SNE for Word Classification task	141
7.10	t-SNE for Behavior Recognition task	142
7.11	t-SNE for Speech Detection task	143

List of Tables

3.1	Number of electrodes by participant for the Single Word experiment.	43
3.2	Number of electrodes by participant for the Harvard Sentences experiment.	45
4.1	Model accuracies by participant for comparison of signal processing methods.	62
4.2	CNN model complexity adjudication	64
5.1	SinIEEG Prediction Success Over Trials	82
5.2	Model Accuracy Comparison	85
6.1	Factor levels for transfer learning experiments	98
6.2	Pretrained model configurations for Participant 1 of the Single Word experiment.	101
6.3	Grand mean of model accuracies across configurations for each participant, compared to the baseline.	103
6.4	Regression to Accuracy of Between-Participant experiment	104
6.5	Regression to Residual Accuracy of Between-Participant experiment	104
6.6	Average accuracy across participants for the Between-Task experiment	108
6.7	Regression for Between-Task transfer learning experiment	110
7.1	Balanced accuracy of downstream tasks. Participant 1 did not have a complete dataset needed for Word Classification and is therefore omitted.	138

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

Alan Turing

1

Introduction

Communicating is an essential part of the human experience. For much of human history, speech communication was primarily intended for other humans but, with advancements in technology, there is an increasing need for communication between humans and computer systems. Further, regardless of target recipient, the majority of all communication is mediated by computer. Whether a keyboard and mouse or speech-to-text, semantic information from language or speech is conveyed to a computer via peripheral devices. A device that decodes intended speech directly from brain activity would be called a speech brain-computer interface (BCI), or speech neuroprosthesis.

The realization of such a device would have broad impact, and could fundamentally change the way humans communicate with computers and potentially each other. It could serve as both a communication device, returning speech to those that have lost the ability; or a control architecture, sending commands to computer systems for control of robotics or IoT homes. Beyond returning function, it could extend it, enabling communication not hindered by the mechanisms of speech, and changing the nature of how we communicate. As with the introduction of any device of such great potential, there is an equal danger of the creation of greater inequity. Therefore, it is critical that in the development of such technologies, great care is taken to ensure that they be accessible to as many people as possible.

There is still significant work to be done in the creation of a practical speech neuroprosthesis, and even more before it is capable of performing outside of a controlled research setting. However, to ensure the goal of equitability in future endeavors, it is imperative to begin embedding universal design principles in the development of speech BCI.

The majority of current speech BCI development has been conducted using invasive BCI designs, decoding speech processes from the electrical signals of intracranial electrodes. However, prior efforts predominantly utilize modular, customized, and data-independent neural feature extraction for the development of decoding models. While this approach has served the field well and produced impressive results, it is suboptimal for models intended for broad application.

Utilized neural features have been largely based on prior human neuroscientific studies with relatively few participants performing narrowly-focused tasks [91]. Participants with intracranial electrodes are often medicated for their condition and palliatively, which has been shown to augment neural processes [33].

Additionally, participants are not a representative cohort, exclusively sampled from a sub-population of people suffering from drug-resistant epilepsy, generally without the types of speech deficits that present research attempts to address. Independent of study samples, inter-person brain morphologies are highly variable, and some degree of variability exists in the neural localization of speech processes [115]. Furthermore, neural dynamics, especially in speech processes, can change throughout the lifetime [168].

Given the large amount of inter-person variation, standardized features derived from studies with such small sample sizes cannot be expected to generalize well to a broader population. Beyond this, there is no single set of neural features that are optimal for the population. In the development of a device with such need for universal access, it is not sufficient to develop it to target efficacy for 95% of people. On the other hand, it is impractical for neural engineers to design a BCI with features tailored to each user. Thus, speech BCIs must be robust to the variability inherent in neural processes, and developed in a way that is person-specific, task-agnostic, and able to evolve over time.

Approach

This work centers around a modeling philosophy that focuses on the engineering of algorithms that automatically learn features from data, rather than the direct engineering of features themselves. Rather than drawing specific conclusions from findings of prior, tangentially-related, neuroscience studies, the proposed approach implicitly identifies information relevant to the speech process as part of a data-driven modeling design. The approach is thus inherently user-specific, with a single parameterized model able to maximize performance for each use-case. Such a design also eliminates the need for any signal pre-processing or conditioning prior to model application. By developing models

that are not contingent on custom or complex signal processing paradigms we lower the barriers to implementation and reproducibility, which is currently a major problem in the field. Such modeling frameworks can also be evolved to expand applicability and efficacy, further accelerating progress.

Deep learning is an empirical modeling method in which the feature extraction and model prediction mechanisms are deliberately combined and inextricable. Over the last decade, the method has proven to be the most effective modeling approach for challenging modeling problems involving unstructured data with complex representations such as computer vision and natural language processing. Unlike other machine learning methods, deep learning performance scales well with data, and model parameters can be iteratively updated and tuned as data evolves over time. Given these model attributes, deep learning is a modeling methodology well-suited for the development of generalized, adaptable speech BCI.

The proposed modeling methodology is evaluated using datasets consisting of intracranial EEG signals collected as participants performed various speech tasks. Electrocorticography (ECoG) and stereotactic electroencephalography (sEEG) are two intracranial BCI modalities that have sufficient spatial resolution and spectral bandwidth to adequately model the complex dynamics of speech processes. Additionally, the majority of existing studies utilize ECoG or sEEG, which provides a corpus of results that allow for direct comparison.

1.1 Organization

This work furthers the goal of developing robust, user-centered speech BCI. Specifically exploring the use of deep learning methods for decoding speech processes. The concepts and results presented in each chapter intentionally

build upon prior chapters to ultimately establish the efficacy and impact of the proposed universal design.

Chapter 2 broadly introduces the topics relevant to this work, including brain-computer interfaces, deep learning, and neuroscience of speech processes, as well as prior work in each domain that this work is influenced by and builds upon. In order to reduce repetition, information about the two datasets used throughout this work is consolidated in Chapter 3. The datasets are covered in detail, including experimental protocols and data collection methods. Chapters 4 through 7 comprise the contributions of this work and are introduced in greater detail here. Finally, Chapter 8 combines threads from the four analyses to summarize the overall results and conclusions of the work. Furthermore, the implications of the results on current modeling paradigms, and potential avenues for continuing future research directions are discussed.

1.1.1 Contributions

Comparing Data-Driven to Preconceived Features

As mentioned, the conventional approach for speech BCI modeling is to use customized, modular, data-independent feature extraction, which also permeates to studies using data-driven approaches such as deep learning. Combining preconceived feature extraction with deep learning is computationally inefficient, and impairs reproducibility. Furthermore, such conditioning methods could be destroying task-relevant information with spurious signal transformations. The contribution of Chapter 4 is to show that, when using a data-driven deep learning modeling approach, explicit signal pre-processing and feature extraction are not necessary. To accomplish this, two competitive feature extraction approaches are compared to an approach using unprocessed signals, with

each method subjected to an equivalent CNN deep learning model. The performance of the three methods is compared on the task of speech activity detection. Following this result, all models and methods in subsequent chapters use unprocessed signals for their analysis.

Learning Person-Specific Features

It is well known that brain dynamics can be highly variable across individuals. While deep learning methods are data-driven, the learned parameters generally have no direct phenomenological interpretation. Chapter 5 shows that deep learning architectures can learn user-specific features, and confirm that researcher-extracted features can exclude useful information for speech modeling. A deep learning model is developed, SincIEEG, which combines mechanisms of CNNs and frequency-domain analysis, a commonly used approach for preconceived feature extraction. In this way, a model is produced which generates data-driven features in the same format as conventional features for direct comparison.

The ideation, design, and implementation of the work supporting Chapter 5 was completed in close collaboration with VCU Department of Computer Science PhD candidate, Morgan Stuart. Morgan's distinct focus and claims in this work are the interpretability and reduced complexity of the resulting engineering-informed model. My distinct focus and claims are the reduced pre-processing and per-user adaptability of what are typically preconceived features.

The models' performance is evaluated on speech activity detection, and several visualization strategies for evolution and comparison of frequency domain features are presented. Results verify previous findings that lower frequency information is useful for speech activity detection, a frequency range commonly

ignored in speech BCI feature extraction paradigms. Further, it is shown that frequency features learned for each participant are varied and unique.

Quantifying the Effect of Transfer Learning for Speech BCI

A common practice to improve training efficiency in the deep learning domain is transfer learning, the use of learned parameters from one model on another. The technique works only if there is a measure of similarity in the data that generate models, in this case, the brain-dynamics of speech processes. Chapter 6 continues the evaluation of SincIEEG, specifically in the context of user-specific features, and quantifying the commonality present in inter-participant brain dynamics.

There has not been a systematic characterization of the effect of transfer learning on speech activity detection from intracranial signals. The contribution of this work is to quantify the effect of model hyperparameters on a transfer learning paradigm. Transfer learning is explored between participants on the same task, as well as within-participant, but on the related speech tasks of overt and imagined speech. The limitations of transfer learning are characterized in the context of speech activity detection, as well as contexts when transfer learning can be effective.

Hidden unit Brain Representations from Intracranial Signals

Several elements have been lacking from current speech BCI modeling approaches. Chapter 7 presents a model and novel implementation that addresses all of them simultaneously. It is well-known that the absolute location of neural activity in reference to the brain is critical information. However, modeling approaches to date, even ones utilizing deep learning, have only employed the relative location

of electrodes. Here, the model is further empowered by converting electrode location information for all participants to a common brain atlas, and embedding this positional information into the model. Prior studies have mapped neural activity to lexical sub-units such as phonemes and words. However, this requires time-intensive labeling of data for supervised training. Our approach overcomes this barrier by utilizing a clustering method that derives self-defined lexical ‘hidden units’ directly from data, without the need for labeling. Along with the difficulty of labeling, a significant problem in the speech BCI domain, particularly using invasive modalities, is a lack of available data. Often, experiments involve a corpus of data on the order of minutes for a participant. Deep learning as a modeling method is notoriously data-dependent.

As with Chapter 5, Chapter 7 also represents a collaboration with Morgan Stuart. Morgan’s distinct focus and claims in this work are the adaptability from unlabeled data using self-supervised learning. My distinct focus and claims are the encoding of spatial information during pretraining and hidden unit representations.

In order to address these constraints, HUBRIS is presented, a transformer-based approach for learning hidden unit speech representations from unlabeled data. HUBRIS employs a self-supervised learning methodology, neuroanatomical positional embeddings, and the contextual representations of transformers to achieve three novelties: (1) learning from unlabeled intracranial brain signals, (2) learning from multiple participants simultaneously, all while (3) utilizing only raw unprocessed data. Furthermore, the speech representations learned by HUBRIS are evaluated using a leave-one-participant-out validation procedure, where weights are transferred to a hold-out participant, and evaluated on three downstream tasks: speech activity detection, speech-related behavior recognition (e.g. listening, speaking, imagining), and word classification.

This analysis represents the first self-supervised transfer learning implementation on intracranial signals, capable of pooling data from an arbitrary number of participants, with the inclusion of absolute anatomical electrode location information.

"I didn't have time to write you a short letter, so I wrote you a long one."

Mark Twain

2

Background

This chapter provides a broad overview of foundational topics relevant to the presented work. First, fundamental neuroanatomy and physiology are introduced and discussed in terms of the neural processes pertaining to speech, the modeling target of this work. Then, BCIs are introduced and signal acquisition discussed in terms of the underlying physiology, in order to motivate the signal characteristics of intracranial EEG and why it is well-suited for modeling speech processes. Finally, deep learning is reviewed and its mechanisms are discussed for the attributes which make it a modeling methodology appropriate for the modeling approach argued for in this work.

2.1 The Brain

The human brain is the most complex organ of the body, and one of the world's most sophisticated parallel distributed information processing systems. Mechanistically, there is no thought we generate or sensation we experience that is not represented by a sequence of a set of neurons firing in the brain. Understanding the structure and mechanism of the brain is fundamental to understanding how it functions. Here, a brief summary is given of neuroanatomy aspects that are pertinent to this work.

Neurons

The neuron is the necessary building block of the nervous system. The carrier of the neural signal. The brain is comprised of a network of approximately 10^{11} neurons, whose function is modulated and mediated by an orchestra of neurotransmitters. These chemical signalers drive the action potential, which ultimately propagates information throughout the brain [47]. Neuronal function is important not only in the context of how it translates to signals acquired by bioamplifiers for modeling, but also as a reference to the biological mechanism that inspired deep learning algorithms.

Though there are several different morphologies of neurons, their mechanism of action remains similar. Broadly, dendrites are the parts of neurons that receive information from other neurons. If chemical signaling from neurotransmitters across dendrites crosses a critical threshold, the neuron membrane depolarizes, beginning the action potential. The action potential is an ion gradient that propagates through the axon to the axon terminals. A synapse is the location where axon terminals of one neuron connect to the dendrites of another. Axon terminals release neurotransmitters into the synapse which then attach to

receptors on the dendrites of downstream neurons. Based on the neurotransmitter, synapses can be excitatory, and add to the critical threshold required for signal propagation, or inhibitory, and subtract from it. In this way, each neuron can be considered a node in a network, with input connections and output connections, functioning in a manner where the sum of its inputs determine whether the neural signal will further propagate to its output nodes. Figure 2.6(A) and (B) respectively show a neuron and a set of neurons with axon terminals and dendrites connected at synapses.

Neuroanatomy

The distribution of neurons in the brain is highly non-homogeneous, with neurons organized into larger-scale structures and sub-networks, whose location and morphology are essential to overall function on a non-local level. Figure 2.1 shows a diagram of the brain as well as common anatomical naming conventions.

Structurally, the brain is organized into a hierarchical layered structure resembling an onion, with two hemispheres and bilateral symmetry. While an oversimplification, outer layers increasingly associate with higher-order processes. The deeper brain structures, such as the brain stem and the cerebellum, control subconscious and reflexive processes like heart rate, blood pressure, and aspects of coordination; while the outer layers of the cerebrum and cerebral cortex, particularly in the frontal lobe, are associated with higher-order cognitive processes such as critical thinking and long-term planning.

The outermost layer of the cerebrum is the cerebral cortex, itself comprised of six layers of densely connected, unmyelinated neurons. The cortex is wrinkled to increase cortical surface area, with ridges called gyri, and fissures called sulci. Beneath the unmyelinated grey matter of the cortex, there is white matter in

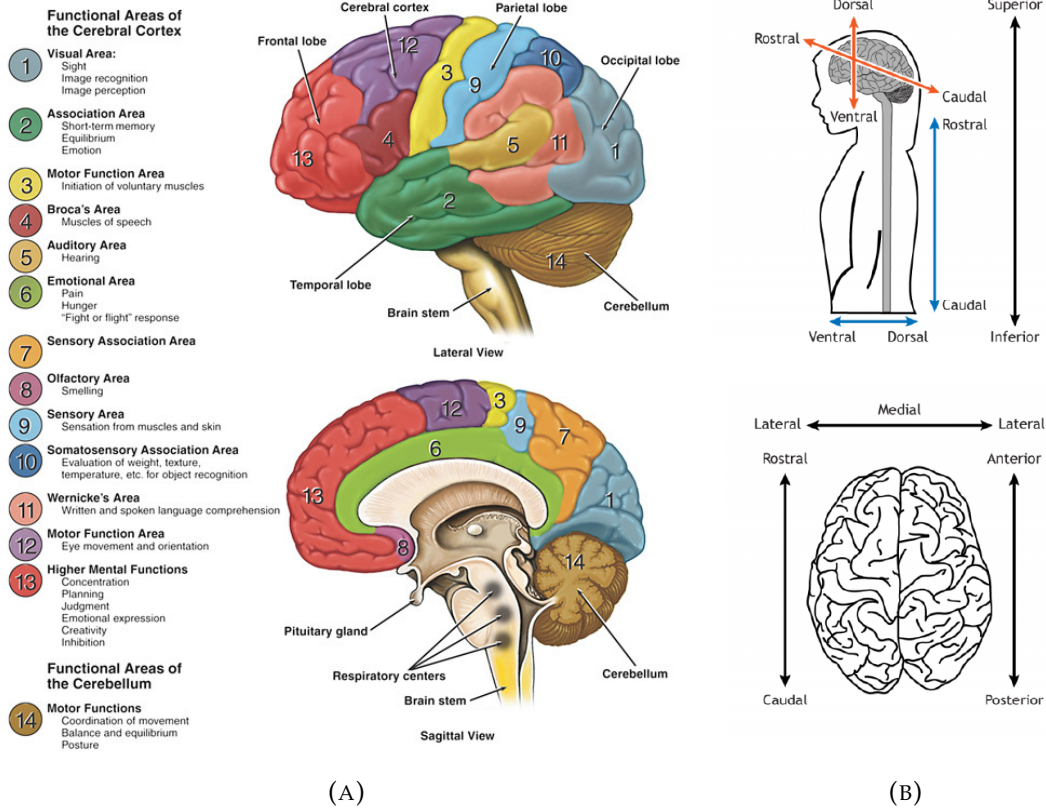


FIGURE 2.1: (A) Diagram of brain structures and functional regions [148]. (B) Anatomical naming conventions used in brain region taxonomy [53].

the cerebrum. White matter is comprised of neurons with myelinated axons, which improve the transmission speed of the neural signal and more efficiently connect distant areas of the brain. In this way, highly localized and complex processing can occur in the grey matter neurons of the cortex, and then the signal is passed to the lower layers of the cerebrum to rapidly propagate the signal a comparatively longer distance [47].

Neural processes often involve highly coordinated activity across many brain structures. A process or related group of processes can be referred to as a neural circuit, and a neural circuit diagram shows the information flow of a given process. Nodes represent brain regions or groups of neurons, and directed edges represent axonal connections. Figure 2.2 shows a diagram of a hypothesized neural circuit for speech perception and production.

The depolarization of neurons, both locally and across broadly connected regions, and the propagation of their signal through the brain structures as well as grey and white matter, synthesize into the low and high frequency oscillatory activity which is sensed by BCI described in the following section.

2.1.1 Speech processes in the Brain

The processes governing speech and speech-related behaviors such as speech planning, overt speech production, inner speech, auditory processing, and language comprehension, are complex, multifaceted, and difficult to tease apart. A great deal remains largely unknown about these processes, as well as other high-level cognitive processes in general.

For example, there is debate on whether language or vocalized speech developed first, or in tandem. While other species, such as members of the primate family, are capable of learning semantic information and the meaning of words,

they cannot vocalize them, implying the former [1]. Further, in both primates and humans, speech-related gesturing has been shown to activate neural circuits for both processing speech and semantic information [38, 161, 166]

Conversely, it has been recently shown that our ability to vocalize, shared by only several other species such as songbirds and parrots, is due to highly specialized neural circuits that allow for the high-speed modulation of our larynx, and has shared genetic origins [25, 165].

The question remains whether some quality of our neural speech circuits, such as their continued plasticity throughout the lifespan, has allowed us to better encode semantic information granting humans an increased capacity for language [18, 94, 168].

Several brain areas have consistently been shown to be associated with specific aspects of speech. Wernicke's area is associated with speech comprehension [160]. The inferior frontal gyrus, which includes Broca's area, is associated with speech production and comprehension as well as language processing [20]. The auditory cortex, located on the superior temporal gyrus, is responsible for integrating sound information from the ear, including speech perception [1, 69, 127]. The primary motor cortex is involved in the articulatory and kinematic aspects of speech production [27, 120]. Speech and language processing are typically lateralized to one dominant hemisphere, most commonly the left, cerebral hemisphere. However, this is not always the case, with 1-5% of right-handed people having a right-dominant language center [81, 149]. In general, there is person-specific variance in the anatomic localization of speech-related brain areas [115].

The method by which the coordination of activity across these critical brain regions gives rise to the various manifestations of speech perception, speech production, and lexical or semantic understanding, is still a topic of debate. Early

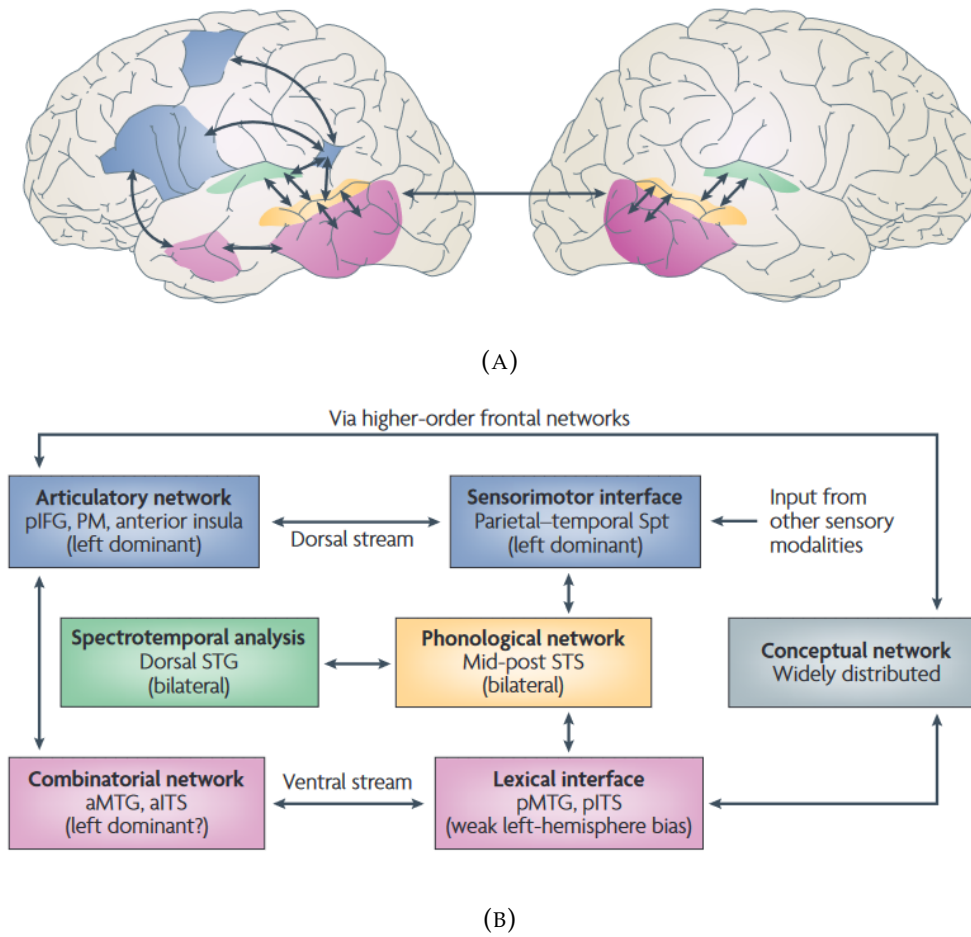


FIGURE 2.2: (A) Diagram of neural circuits involved in speech planning and production. (B) Information flow diagram of the neural circuits corresponding to color-matched to diagram (A). Adapted from [60]

models of speech processing including the foundational Wernicke-Lichtheim-Geschwind model asserted that speech sub-processes had associated neural localizations [95, 107]. Recent models have abandoned this trend in favor of a parallel distributed processing paradigm, wherein multiple sub-processes involving disparate brain regions are in constant concert, with interacting loops involving perception, integration, and production [59, 60, 73, 92].

2.2 Brain-Computer Interfaces

A BCI is a computer peripheral that interacts directly with neural signals generated by brain matter [164]. BCI research has spanned a wide range of applications [71], including rehabilitation after injury such as traumatic brain injury or stroke [29, 85]; clinical and peri-operative patient monitoring [142]; neurofeedback and measurement of cognitive states such as attention and workload [57, 152]; control architectures for wheelchairs, robotics, and prosthetics [62, 65, 138]; communication modalities such as spellers and speech neuroprosthetics [55, 86, 100]. In addition, BCIs are the primary mechanism of collecting data for in-vivo human neuroscience studies; furthering our understanding of how the human brain functions.

While the majority of BCIs are designed for passive sensing neural activity, bi-directional BCIs have been proposed to also stimulate activity. These stimulation technologies include transcranial direct current stimulation (tDCS) and transcranial magnetic stimulation (TMS), as well as deep brain stimulators approved for the treatment of Parkinson's and major depressive disorder (MDD). This work focuses on BCIs that sense neural activity.

2.2.1 Properties of BCI Sensing Modalities

BCIs can be differentiated along several design dimensions, which impact the application contexts.

There are several mechanisms for sensing the activity of neurons. They can be divided into two general groups. One is modalities that sense brain metabolites, such as blood flow, to infer neural activity, including functional MRI (fMRI) [114] or functional near-infrared spectroscopy (fNIRS) [57]. The other group is comprised of modalities that sense the changes in the electromagnetic field

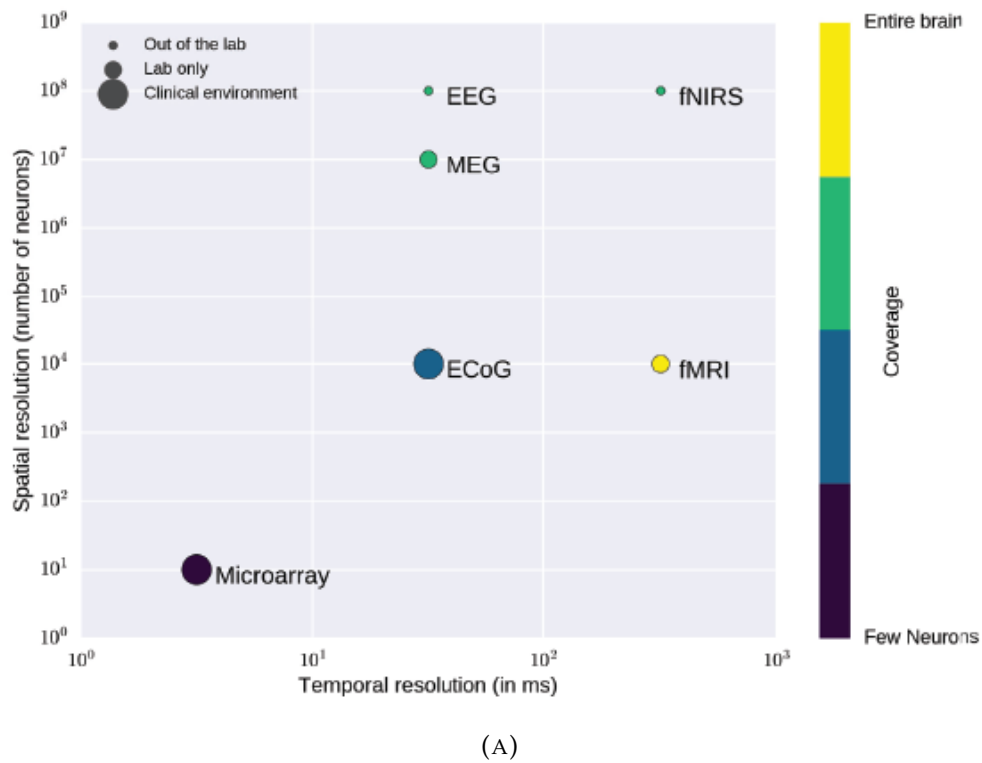


FIGURE 2.3: Signal properties of selected BCI sensing modalities, organized across several dimensions: Temporal resolution (x-axis); spatial resolution (y-axis); portability, with larger circles corresponding to modalities that have significant barriers to functioning in an open world environment; brain coverage, with lighter colors corresponding to modalities able to acquire signals spanning a larger brain area [58]

caused by the ion flux of neuronal depolarization. These include any electrode-based modality, including scalp EEG, electrocorticography (ECoG), stereotactic EEG (sEEG), and microelectrode arrays. Generally, metabolite-sensing BCI are capable of good spatial resolution, but comparatively poor temporal resolution. fMRI in particular is unique in its ability to capture activity from deep brain structures in a non-invasive manner.

A BCI is termed non-invasive if its application does not require a surgical procedure. Scalp EEG, MEG, fNIRS, and fMRI, are all such non-invasive BCI. In contrast, sEEG, ECoG, and microelectrode arrays are invasive BCI. Such BCI are capable of superior signal-to-noise ratios, frequency domain resolution, and depending on modality, near single-neuron spatial resolution. However, this signal quality comes at the cost of the requirement of risky neurosurgery procedures.

Figure 2.3 compares the signal properties of various BCI modalities along the dimensions of spatial and temporal resolution, coverage, and appropriate setting.

Measuring Neuroelectrical Signals

Electroencephalography (EEG) refers to BCI which measure changes in the local electric field via electrodes. When a neuron depolarizes and an action potential begins, a dipole moment is created in the surrounding electric field, oriented approximately in the direction of its axon. This dipole moment can be conceptualized as a vector in 3D space. Each neuron that fires creates this dipole, oriented along the axis of its axon. The vector dipoles of all surrounding neuronal activity are summed into an aggregate dipole moment.

BCI that sense electrical activity are measuring this dipole moment. In the case of all electrode-based BCI, this equates to measuring the aggregate dipole

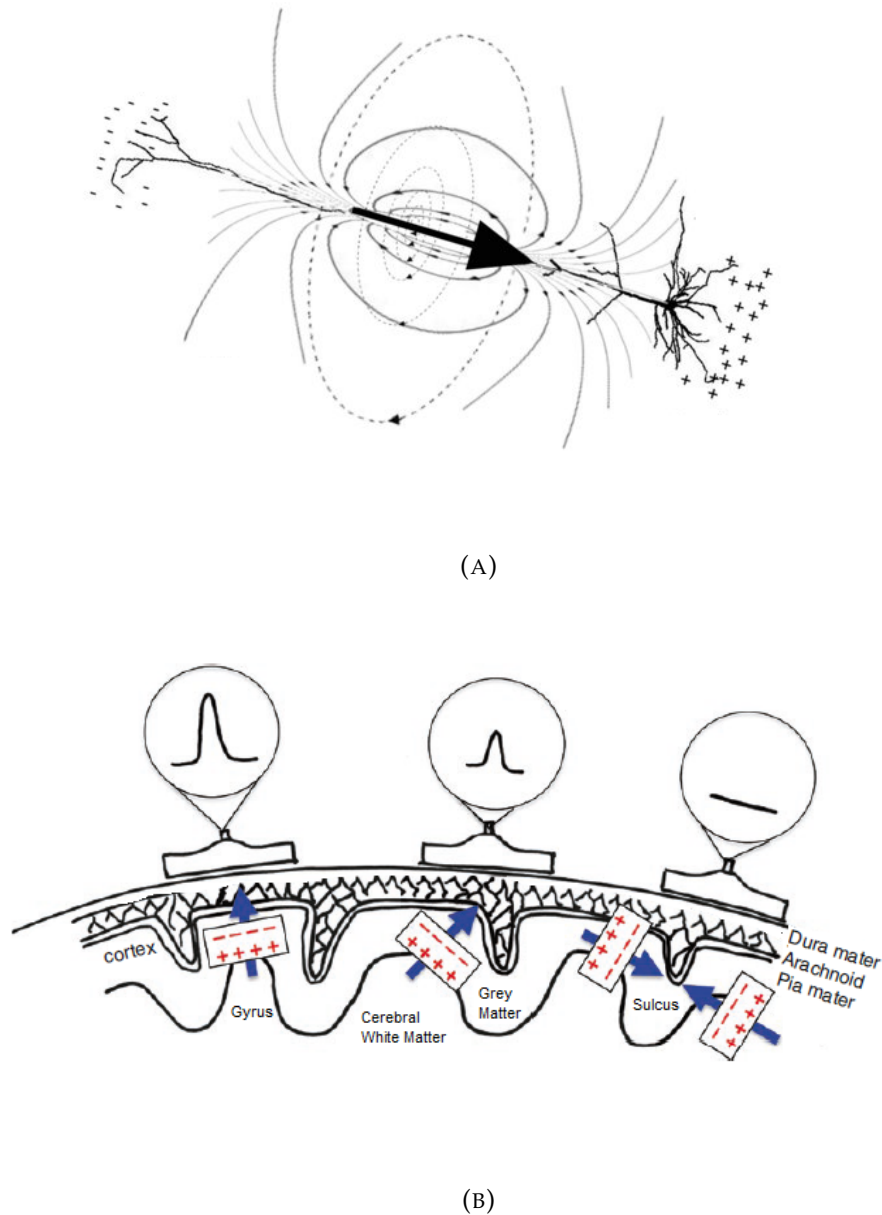


FIGURE 2.4: (A) Diagram of a neuron depolarizing, and the action potential producing a dipole moment in the electric field. The flux of the dipole in turn produces a change in the magnetic field. (B) A diagram of epidural ECoG electrodes showing how placement above gyri or sulci can produce different measurements due to constructive or destructive dipole aggregation.

projected onto the hyperplane between the measuring and reference electrodes. In the case of MEG, the apparatus is instead measuring the flux in the magnetic field that is generated by the appearance of the dipole moment.

Signals normally dissipate at a square distance rate, and this phenomenon inherently creates a trade-off. Electrodes closer to neural activity will receive greater signal amplitude from nearby neurons. Thereby, they can either measure the precise activity of a small set of neurons, such as in the case of sEEG, ECoG, or microarrays; or the combined activity of larger brain regions at the cost of spatial resolution, such as with scalp EEG.

2.2.2 Stereotactic EEG and Electrocorticography

The present work focuses on the development of decoding models using intracranial EEG (iEEG) signals, specifically sEEG and ECoG. In the clinical setting, both modalities are commonly used as part of the procedure for the localization of seizure activity for patients suffering from intractable, drug-resistant epilepsy.

In the case of scalp EEG, biopotentials from neurons must travel through the dura, skull, and scalp in order to reach the electrode. This non-active organic material serves to diffuse the neural signal and acts as a low-pass filter. Thus, EEG is best suited to measure the slower oscillatory activity of neurons across comparatively larger brain regions. Because intracranial EEG electrodes rest directly against neural tissue, or on the dura, they do not suffer from these same drawbacks and are capable of capturing high-gamma band information from the frequency domain [15].

Intracranial EEG signals possess high spatial resolution (cm to mm) and very high (ms) temporal resolution. Additionally, because they are implanted,

the signals are not contaminated by movement artifacts or electromyographic (EMG) artifacts from ocular or scalp muscle contractions.

These signal attributes make sEEG and ECoG ideal choices for the exploration of complex and distributed processes such as speech and language. While ECoG provides excellent coverage of the cortex important for modeling the distributed processes, it is generally unilateral, and unable to capture information from deeper brain structures. Conversely, while sEEG is able commonly implanted bi-laterally and can access subcortical activity, because shafts are oriented normally to the skull surface, cortical coverage is comparatively sparse.

In this way, the two modalities provide complementary data about the brain during speech processes.

Electrocorticography (ECoG)

Electrocorticography (ECoG) is an intracranial EEG in the form of flat electrode grids. These grids are placed directly on the brain surface, either beneath or on top of the dura mater, and require a craniotomy to place [122, 139].

Electrodes measure activity from the cortex, highly preferential to the activity of neural tissue directly under each electrode. As referenced in Sections 2.2.1 and 2.1, the pyramidal neurons of the cortex are oriented normal to the cortex surface, and the dipole moment will also be oriented in this direction. In sulci the cortex folds inward, producing a dipole oriented parallel to the neurons located on a gyrus. This can result in neural activity appearing differently if the measuring electrode is located on a gyrus or sulcus.

ECoG electrodes are comprised of platinum iridium with a diameter of 2-4 mm. Inter-electrode distance for grids can vary between 1 cm to 1.25 mm for high-density grids [144, 156]. Generally, non-high-density grids are comprised of 16-64 electrodes. Each electrode signal is acquired by the bioamplifier, and

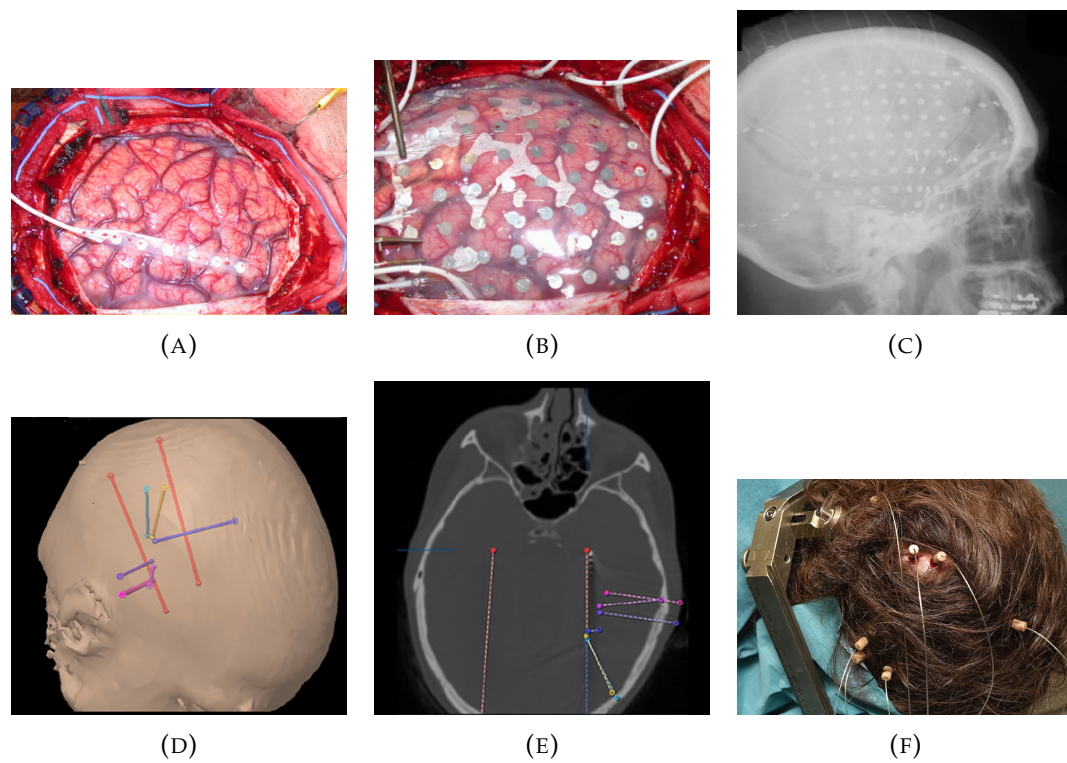


FIGURE 2.5: (A) Exposed brain surface. (B) ECoG grid perioperative placement. (C) Post operative x-ray showing electrode locations. Adapted from [91] (D) Trajectory planning for sEEG shafts. (E) CT of implanted sEEG electrodes. (F) Head post implantation. Adapted from [54].

represents one channel of neural data. Though it varies by clinical need, there are commonly 32-96 electrodes per patient. Figure 2.5 (A)-(C) show an example of pre-, peri-, and post-operative ECoG implantation.

Though most participants in research experiments are implanted for 1-4 weeks, it has been shown that the ECoG recordings from chronically implanted participants do not degrade over a period of up to 2 years [113]. Particularly epidural placement does not illicit the same neural scarring and electrode signal degradation that can be present in sEEG.

Stereotactic EEG (sEEG)

Stereotactic electroencephalography (sEEG) is an iEEG modality that takes the form of shafts, with electrodes placed along them, [16]. The shafts are positioned via stereotactic guidance, and implanted into the brain via burr holes in the skull. Although the surgical procedure does not require a craniotomy as ECoG does, it is not without risk, and careful planning of shaft implantation angles and penetration trajectories is required to avoid rupturing blood vessels [28].

In contrast to ECoG, which is capable of providing excellent coverage of large areas of the cortex, sEEG has comparatively little cortical coverage. However, the depth-wise nature of the modality allows for access to deeper brain structures such as the hippocampus, thalamus, and basal ganglia, as well as the white matter tracts of the subcortical cerebrum [93, 132].

In general, there are 8-18 per shaft, with 1.5-3.5 mm between electrodes. Commonly there are 5-10 shafts per patient, for generally the same order of magnitude channels as there are for patients implanted with ECoG grids [97]. sEEG shafts are similar to those used for Deep Brain Stimulation (DBS), a common treatment for drug-resistant Parkinson's Disease and other movement disorders. As such, there is ample data on the effect of long-term implantation of sEEG [54,

96]. Figure 2.5 (D)-(F) show the pre-operative planning, and post-operative CT and implanted shafts for a sEEG surgical procedure.

2.2.3 Speech Processes from Invasive BCIs

Over the past decade, there have been great strides in the development of neural speech prosthetics from invasive BCI [26]. Early attempts focused on further investigating features and functional organization of speech perception and production posited by the models developed in previous decades [60, 92].

One ECoG study analyzed the cortical activity over the ventral sensorimotor cortex, and confirmed the areas involvement in vocal articulatory kinematics, mainly via principle component analysis [19]. Another study by the group showed early reconstruction of speech elements from ECoG signals from the auditory cortex [116]. In a study on the temporal dynamics of speech, the group showed that activity in the speech planning Broca's area preceded activity in the auditory and motor cortex, and that high gamma frequencies held the most relevant information for speech, a finding that would inform many foundational feature extraction schemes [91].

Following studies largely geared towards characterization, subsequent efforts focused on decoding aspects of speech or speech primitives. Phonemes were classified with 36% accuracy from ECoG signals of the motor cortex using broadband gamma features [112]. Other studies focused on characterizing ECoG features of continuous speech and inner speech [98, 102].

Other approaches attempted to classify words from neural signals. One study using broadband gamma features and support vector machines classified individual words from both overt and imagined speech [103]. Two other studies showed textual decoding of continuous speech using language models and

approaches from the field of Automatic Speech Recognition [55, 109].

Instead of decoding speech primitives or words, other efforts showed that single word speech waveforms could be directly synthesized from ECoG signals [58]. This was later extended to a real-time ready approach for continuous speech synthesis [56]. A subsequent study showed that the real-time synthesis of imagined speech was possible[6] using the same techniques, which involved a unit selection modeling approach and broadband gamma envelope features.

The path of research has been from characterization to implementation, from decoding overt speech to imagined speech, and from offline models to models capable of online use. There are common elements across this corpus of work.

Nearly all of the aforementioned analyses use preconceived features from broadband gamma frequencies, and it is evident that they are capable of at least partly modeling speech processes, both overt and imagined. Most studies have used ECoG datasets, and consistently shown the parallel and distributed nature of the speech process, and that different brain regions can be utilized to model speech. In terms of timing, speech is a relatively temporally localized process, with the majority of the speech information process, from planning to execution and perception, occurring within 400-600 ms of word utterance. Commonly, a small subset of electrodes were correlated with labeled phenomenon, but in a highly participant-dependant manner. Additionally, these studies exclusively employed supervised learning paradigms. Finally, while simpler modeling approaches like logistic regression, linear discriminant analysis, or support vector machines, were capable of performing above chance accuracy, more sophisticated models have been required for more generalized and improved performance.

2.3 Deep Learning

Deep learning (DL) is a subdomain of Artificial Neural Networks (ANN), which are a subdomain of machine learning (ML). The defining trait of machine learning algorithms is that they are *trained* on data, meaning they augment their function in response to that data. Artificial neural networks, or simply neural networks, are a type of machine learning algorithm intended to mimic the biological neuron and brain function described in Section 2.1.

The precursors to neural networks were originally proposed in the 1940s by neuroscientist Warren McCulloch and mathematician Walter Pitts to mathematically model the information processing of neurons [105], and the first example of a neural network model was the perceptron [134]. Like the biological process on which it is based, the elegance of the technique is in simple building blocks that, when combined, gain arbitrary modeling complexity.

A brief summary of the fundamental neural networks is given, and to motivate the concept a comparison is drawn to the biological inspiration already introduced in Section 2.1. Figure 2.6 shows a biological neuron with axons from other neurons connected to its dendrites, and the diagram for its digital counterpart.

As is the case with its biological counterpart, the artificial neuron performs a weighted summation of its inputs. Then, the weights are passed through a non-linear function, the activation function. This is a continuous function on the space of real numbers, that reduces values below zero to zero and amplifies values above zero. In this way, it simulates the thresholding required for a biological neuron to reach the voltage necessary to depolarize and the action potential to propagate. For ANNs, neurons are the functional unit, organized into layers, with outputs from preceding layers serving as inputs to following layers.

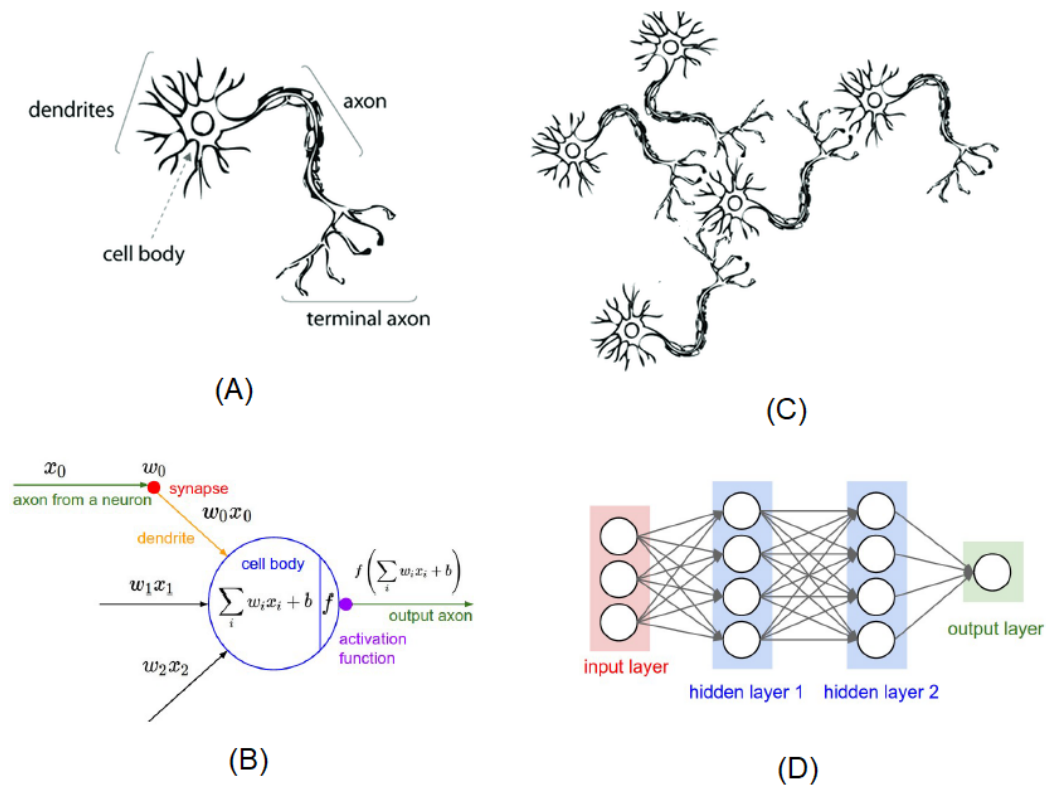


FIGURE 2.6: (A) Diagram of a single neuron. Including dendrites, where signals are input, the cell body where the signals are aggregated, and the axon where the action potential propagates. (B) An artificial neuron. Weighted signals are summed, passed through an activation function, f , and serve as inputs to downstream neurons. (C) A network of biological neurons, chemical signaling from axon terminals can be either excitatory (encouraging action potential) or inhibitory (discouraging action potential). (D) A fully connected ANN with two hidden layers. Weights close to 0 simulate effectively disconnected neurons, negative weights are analogous to inhibitory signals, and positive weights to excitatory signals.

Layers other than the initial input layer and final output layer are called *hidden layers*. A fully connected ANN with a single linear layer is a universal function approximator. ANNs with more than one hidden layer are considered deep neural networks (DNNs), and machine learning algorithms employing deep neural networks are thus termed deep learning. [64].

A note on taxonomy

The terms machine learning, deep learning, deep neural network, artificial neural network, and neural network, have all already been used here. The deep learning domain already suffers from nuanced naming conventions. Additionally, this work is at the intersection of that domain and the underlying phenomenon on which it is modeled. In order to avoid confusion, several naming conventions used throughout the work are stated explicitly.

As contemporary model complexity has increased, most models have evolved to have many hidden layers, and as such most networks would be considered deep neural networks. This is the case for all model applications in this work. The terms deep learning, neural networks, or simply networks, are synonymous and used interchangeably unless explicitly stated.

Importantly, a distinction is made between the artificial and biological domains with keywords. We refrain from using the term *neuronal* network, sometimes used to describe a network of biological neurons, because of the similarity to *neural* network. Instead, the term neural *network* is used in this work to refer to the machine learning algorithms, while the terms neural *circuit* and neural *process* refer to biological brain functions.

2.3.1 Supervised Learning

Supervised learning is not a subset of deep learning, but rather a type of machine learning algorithm. It is covered here as it is the type of algorithm employed in the majority of this work, as well as all of the studies referenced in this chapter.

Supervised learning algorithms are those where the input to the model during training is paired with a label representing the ground truth output corresponding to that input. Each data point is an input-label pair. The output prediction of the model for a given input can then be compared to the label, thus the labels supervise the training procedure. The learning algorithm attempts to find the most accurate mapping from the inputs to the labels.

These labels can be a categorical variable, taking one of an arbitrary number of classes. This type of predictive problem is called classification. If the labels are a continuous or ordinal variable, this type of problem is termed regression.

Training and Evaluation

In order to effectively train the model and evaluate its performance, data is split into training, cross-validation, and testing sets. First, a portion of the dataset, referred to as the test set, is withheld entirely from the training procedure in order to evaluate how well the model generalizes performance to previously unseen data. With the remaining data, a majority portion is used for training the model, while the rest is used for cross-validation. This is a model validation measure that is employed during training and serves as an estimate of training-complete model performance. It can be thought of as a checkpoint in the training process [51].

2.3.2 Deep Learning Model Design and Training

To reiterate, deep learning is an empirical modeling method, meaning that the evolution of model parameters is driven by the data input to the model. In order for a deep learning model to learn, several components are necessary.

Loss Function.

The loss function, sometimes referred to as the objective function or criterion, is how the model quantifies quality of the prediction, \hat{y} for a given input, x . An entire neural network can be thought of as a compound function, $f: X \rightarrow Y$, which maps the set of inputs X to the set of labels Y . The loss function is a function, $L(\hat{y}, y)$, that measures the difference between predicted \hat{y} and the true label y . Even in the simple case where $y \in \mathbb{R}$, there are choices for L . For example, $L(\hat{y}, y) = |\hat{y} - y|$, the absolute value or L_1 norm; or $L(\hat{y}, y) = (\hat{y} - y)^2$, the L_2 norm or square distance, are both valid choices.

The choice of loss function will impact the model drastically as, in the multi-dimensional sense, it creates the topology of the problem space.

Optimizer

If the loss function represents the topology, then the optimizer seeks to find the lowest point possible in the problem space. To continue the analogy and summarize a many technical details, the choice of optimizer will determine the direction and size of the next step taken towards a local minimum, or ‘valley’, in the problem space. Stochastic Gradient Descent (SGD) is likely the most well-known optimization function. However, in recent years there have been improvements in optimizers that do not suffer from the same drawbacks as SGD,

namely noisy steps and slower convergence rates. In this work, the Adam optimizer is used for all analyses, which has quickly become a standard in the deep learning field [80].

Backpropagation

After the optimizer calculates the step to take, the weights of the network must be updated accordingly. Like water carving a path into a mountainside, it is this updating procedure that evolves the network into a well-suited mapping of the inputs to the outputs. The process of backpropagation is what makes deep learning such a powerful modeling methodology. Because all neuron units are a linear combination of weights and inputs, and all activation functions are, by design, fully differentiable, the network at any node can be conceptualized as a nested function. Backpropagation is the recursive application of the chain rule on that function. The updating of the weights at each layer depends only on the gradients of the the layer after it. The first layer to be updated is the last layer in the network, and updating propagates backwards layer-wise throughout the network towards the initial layers, hence backpropagation [48, 79].

Network Architectures and Activation Functions

Ultimately, the network architecture has the most bearing on modeling function and capacity. An important element of the network neurons is the choice of their activation function. Activation functions effect how inputs propagate through the network, and can effect training pitfalls such as exploding gradients. To apply backpropagation, activation functions must be differentiable and defined across the set of real numbers. Activation functions are not required to be consistent across network layers, but it is common that they are. Several common

activation functions are shown in Figure 2.8. The rectified linear unit (ReLU) function in particular has been instrumental in recent deep learning innovations [2].

Figure 2.7 shows three common network architectures. The first is a traditional neural network, and the architecture that has been referenced in this section thus far, called a *feed-forward network*. It is fully connected, meaning that neurons from one layer are connected to all neurons on adjacent layers. These architectures are still used, but often as components of more complex networks.

The middle architecture is a *convolutional neural network*, commonly used in computer vision tasks, and the network architecture responsible for the resurgence of interest in deep learning a decade ago [136]. This type of network is used throughout this work and is discussed in greater detail in Chapter 4.

Finally, the last architecture is a *recurrent neural network* (RNN). Expanding upon a feed-forward network, an RNN allows for the modeling of temporal or sequence behavior by feeding the output of nodes to act as inputs to the same nodes in the subsequent time step. Following the success of CNNs for computer vision, RNNs and sequence models provided similar advancements in the field of natural language processing. RNN architectures were the precursors to current language models.

The field of deep learning in its current form is relatively young and rapidly evolving. While implementations are well-established for tasks like computer vision and natural language processing, the field of deep learning for BCI is still in its infancy.

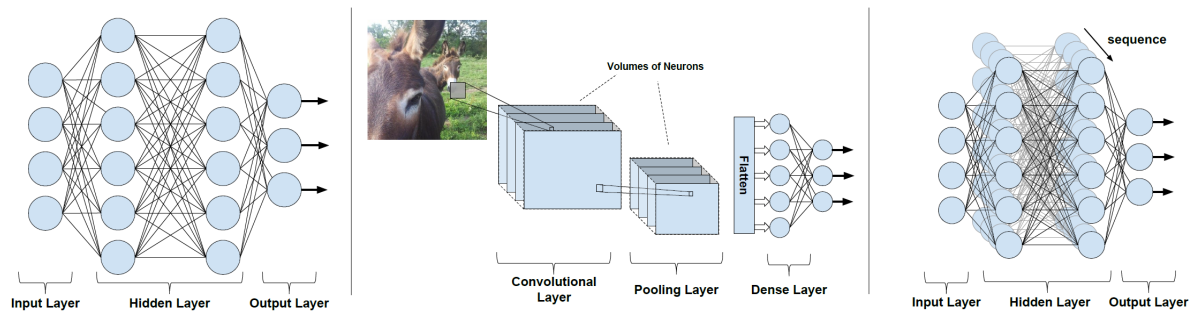


FIGURE 2.7: (Left) A fully connected feed-forward neural network with two hidden layers. (Middle) A convolutional neural network (CNN) with a convolutional layer, a pooling layer, and a fully connected output layer. (Right) A recurrent neural network version of the feed-forward network on the right. The network's previous state are inputs to the current state.

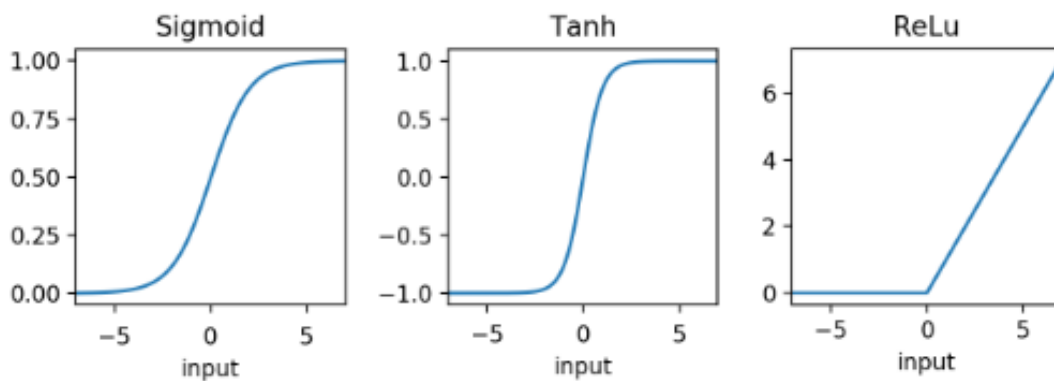


FIGURE 2.8: Common activation functions. (Left) The sigmoid function, commonly used in output layers. (Middle) Tanh function, a common alternative to sigmoid for binary classification. (Right) The Rectified Linear Unit (ReLU) function, common throughout deep learning architectures.

2.3.3 Deep Learning for Speech BCI

Applications of deep learning to decoding speech processes are relatively new to the field, with successful early efforts emerging in 2019. The studies can be split into two groups; those attempting to synthesize speech, and those attempting to decode words or sentences as textual representations.

One ECoG study showed speech waveforms could be successfully synthesized using relatively simple architectures comprised of convolutional neural networks and fully-connected feed-forward networks [3]. The study compared the deep learning approach to linear regression methods, with and without a vocoder to synthesize the audio waveform. Another ECoG study used 3D densely connected convolutional neural networks to synthesize speech, also reconstructing Mel Frequency Cepstral Coefficients (MFCCs) and using a vocoder to construct the audio waveform [7].

Departing from CNNs, a study using recurrent neural networks for the synthesis of spoken sentences by modeling articulatory kinematics [8]. Continuing with the trend of reconstructing MFCCs, the study used a two-stage bi-directional Long Short-term Memory (LSTM) network. A recent study, and the only study to use sEEG data, has shown impressive results by making use of a state-of-the-art text-to-speech model, the Tacotron-2. Instead of text embeddings, a CNN is used as a feature encoder. MFCCs were reconstructed, and the WaveGlow vocoder used to reconstruct the audio waveform [82]. The Tacotron is a bi-directional RNN-based architecture. While the results for speech synthesis have improved, there has not yet been an approach that has been able to consistently produce intelligible-quality speech.

Two studies from the same group focused instead on decoding textual representations. A study using RNNs and a sequence-to-sequence encoder-decoder

architecture was able to achieve a 3% word error rate [100]. The study performed better when the encoder used MFCC reconstruction as an objective of the encoder during training.

Finally, representing the state of the art for speech neuroprosthetics, a study was able to achieve online, near real-time decoding of speech for a single patient with anarthria using a hierarchical ensemble of deep learning models and language models [110]. The decoding model has several stages. First, a speech activity detection model identifies when the patient is intending to speak, using an LSTM architecture. Sections of ECoG signals identified as intended speech are passed down to a word classification model, of a similar structure to the one presented in [100], which classified words from a restricted lexicon. Finally, a Viterbi language model calculates the most likely sentences as words are classified. Classification performance, even on a reduced set of words, still requires improvement. However, to date, this is the only study to show online decoding of speech processes.

For both speech synthesis and text decoding, deep learning methods have produced the state of the art results. However, with the exception of one study [82], all studies presented employed researcher-derived features. Moreover, studies used only supervised learning methods, with models tailored to participant sets. There is still progress that must be made in order to produce robust models regardless of the task application.

*"The brain doesn't ask 'What is this?'. The brain asks
'What is this **like**?'"*

Lisa Feldman Barrett

3

Intracranial EEG Datasets

The analyses and modeling presented in Chapter 4 through 7 of this work are carried out on two datasets. Here, the datasets are introduced, along with detailed experimental protocols, participant electrode coverage, and data collection methodologies. Beyond this, any analysis-specific data manipulation, such as sample labeling schemes, are covered in their respective chapters.

3.1 Single Word Experiment - Electrocorticography

Experimental Protocol

Participants were instructed to read aloud single words presented in sequence on a computer screen while their brain activity and voice were simultaneously recorded. Figure 3.1 shows a diagram of the experiment. The words were selected from a bank of 431 unique words, split into 4 sets of 115-116 words. The bank of words are primarily monosyllabic and comprised of the Modified Rhyme Test [66], supplemented with additional words to better reflect the phoneme distribution of American English [108]. While this experimental paradigm was originally designed to examine neural correlates of American English phonemes [112], the data are being used in the present analysis exclusively for speech activity detection without consideration of phonetic aspects.

The experiment begins with a fixation cross at the center of the screen. The cross is then replaced by a word that stays on the screen for 2.5 seconds. The

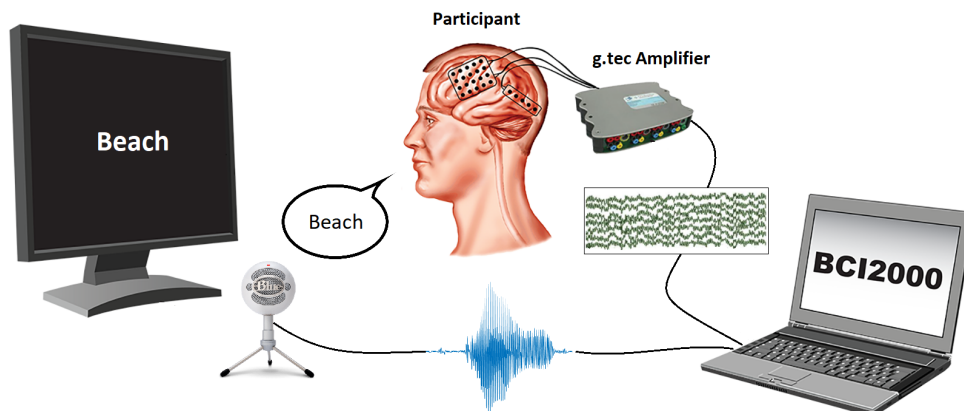


FIGURE 3.1: Diagram of the Single Word experiment. Time synchronized audio and ECoG recordings are recorded while a participant reads aloud words presented on a monitor.

word is then replaced with the cross for 0.5 seconds, before the next word is presented. Words are chosen randomly from the set of 115 words for each session and each session contained different subset of words. Participants completed between 2 and 4 sessions, depending on willingness and ability to complete the sessions.

Participants

ECoG data were recorded from 5 participants with pharmaco-resistant epilepsy undergoing clinical monitoring for surgical planning. No participants reported hearing deficits. In all cases, a tumor was not the source for the seizures and no lesions were indicated by any electrode used for analysis. All participants gave written informed consent and the study protocol was approved by the institutional review boards of Virginia Commonwealth University; University of California, San Diego; Old Dominion University; and Mayo Clinic, Florida.

Participants were implanted with subdural electrode grids or strips (Ad-Tech Medical Instrument Corporation, 1-cm spacing) based purely on their clinical need. Electrode locations were verified by co-registering preoperative MRI and postoperative computerized tomography scans. For combined visualization, electrode locations were projected to common Talairach space. Electrode locations were rendered using NeuralAct [87], as shown in Figure 3.2. While brain areas associated with speech are predominantly found on the dominant hemisphere, which is the left hemisphere in the majority of right-hand dominant people, the neural correlates of speech production are not exclusively localized in the left hemisphere [34, 118]. For this reason, both left and right hemisphere cases are evaluated. In total, ECoG activity was recorded from 416 (96 left hemisphere, 320 right hemisphere) subdural electrodes. Of these, electrodes

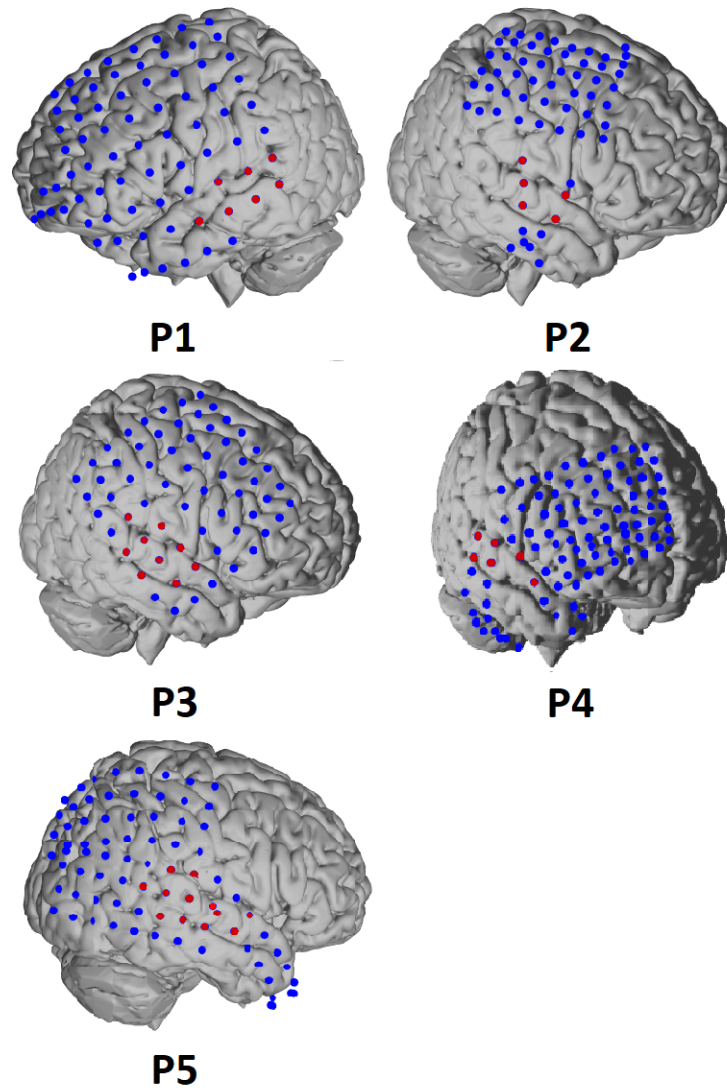


FIGURE 3.2: Electrode locations for all 5 participants. Electrodes identified in the auditory cortex region are highlighted in red.

that exhibited unnatural signal anomalies based on visual inspection were excluded from the analysis, leaving 364 electrodes (96 left hemisphere, 268 right hemisphere). For each participant, the number of electrodes implanted, analyzed, and identified as not located over the auditory cortex (non-auditory) are provided in Table 3.1.

TABLE 3.1: Number of electrodes by participant for the Single Word experiment.

Participant	Implanted	Analyzed	Non-Auditory
1	96	96	89
2	64	51	49
3	64	55	48
4	96	77	73
5	96	85	75
Total	416	364	334

Data Acquisition

ECoG and audio data were concurrently recorded during the task. ECoG data were bandpass filtered between 0.5 and 500 Hz, notch filtered at 60 Hz, and recorded using g.USB amplifiers (g.tec Medical Engineering). The data were recorded at a sampling rate of 1200 Hz and subsequently decimated to 600 Hz.

The time series and its frequency spectra were visually inspected for anomalies. Channels having uncharacteristic frequency spectra, substantial artifacts, and/or saturated amplitudes, were excluded from analysis. In total, 364 (96 left hemisphere, 268 right hemisphere) electrodes were used for analysis.

This basic preprocessing is standard for ECoG acquisition and the data decimation can be equivalently achieved by using a lower sampling rate at the time of data acquisition. Thus, the data used as input to the SincIEEG network effectively represent the raw ECoG timesamples.

Audio data were recorded in parallel using a Blue Microphones Snowball iCE USB microphone connected to the research computer, sampled at 48 kHz. All data recording and stimulus presentation were facilitated by BCI2000 software [137].

3.2 Harvard Sentences Experiment - Stereotactic EEG

Experiment Protocol

The experimental protocol is designed to investigate overt and imagined speech processes in the brain by having participants repeat a sequence of sentences, each in a series of three different speaking modes. Before the experiment, the participant is explained the paradigm, and experimental icons and cues, and is instructed to perform the associated tasks immediately upon cue presentation - within a 4-second interval during which the task cue was displayed.

A trial begins with a short sentence displayed on a computer monitor while simultaneously narrated through computer speakers. All sentence audio was less than 4 seconds in length, but regardless of the length, the associated text remains on the screen for 4 seconds. Following a 20 ms blank screen, the participant is cued with an icon to vocalize the sentence (i.e., overt mode), and this cue remains on the screen for 4 seconds. Following a 20 ms blank screen, the participant is cued for 4 seconds by an icon to articulate the sentence as if they were speaking, but without vocalizing (i.e., mouthing mode). Finally, after a 20 ms

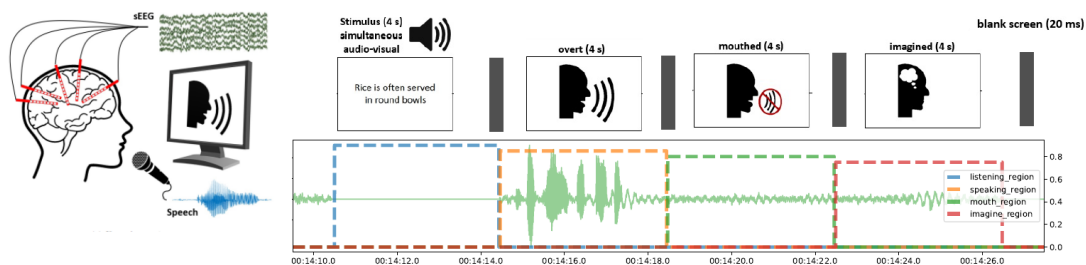


FIGURE 3.3: Harvard Sentences experiment protocol diagram. The right of the diagram represents the participant regarding the experiment cues on the screen and performing the corresponding task, while audio and sEEG signals are recorded. The right shows an example of the display cues during an experiment trial, and the corresponding audio signal below.

blank screen, the participant is cued for 4 seconds by an icon to imagine speaking the sentence without articulating or vocalizing (i.e., imagined mode). Then following a 20 ms blank screen, the next sentence trial begins. This protocol is illustrated in Figure 3.3.

The paradigm is repeated each time for a set of 50 unique Harvard sentences, designed to be phonetically-balanced conversational English [135]. All participants completed the entire set of 50 sentence trials; however, only 25 sentence trials from Participant 1 are evaluated due to a software issue that corrupted the labeling of the other 25 sentence trials.

Participants

sEEG data were collected from 7 native English-speaking participants being monitored as part of treatment for intractable epilepsy at the University of California San Diego Health. The locations of sEEG electrodes were determined solely based on the participants' clinical needs. The number of electrodes implanted for each participant is provided in Table 3.2. The study was approved by Virginia Commonwealth University and UCSD Health IRB.

Participant	# Electrodes
1	90
2	70
3	80
4	175
6	94
7	108

TABLE 3.2: Number of electrodes by participant for the Harvard Sentences experiment.

Data Acquisition

Data from the sEEG electrodes (Ad-Tech Medical Instrument Corporation) were recorded with a Natus Quantum Amplifier(Natus Medical Inc.) and referenced to a pair of subdermal needle electrodes in the scalp. The amplifier signals were digitized at 1,024 Hz. An external microphone recorded the audio signal, which was digitized at 44,100 Hz. The digitized intracranial signals and microphone audio, along with the experiment cues, were synchronized with the Presentation® software (Version 18.0, Neurobehavioral Systems, Inc., Berkeley, CA, www.neurobs.com).

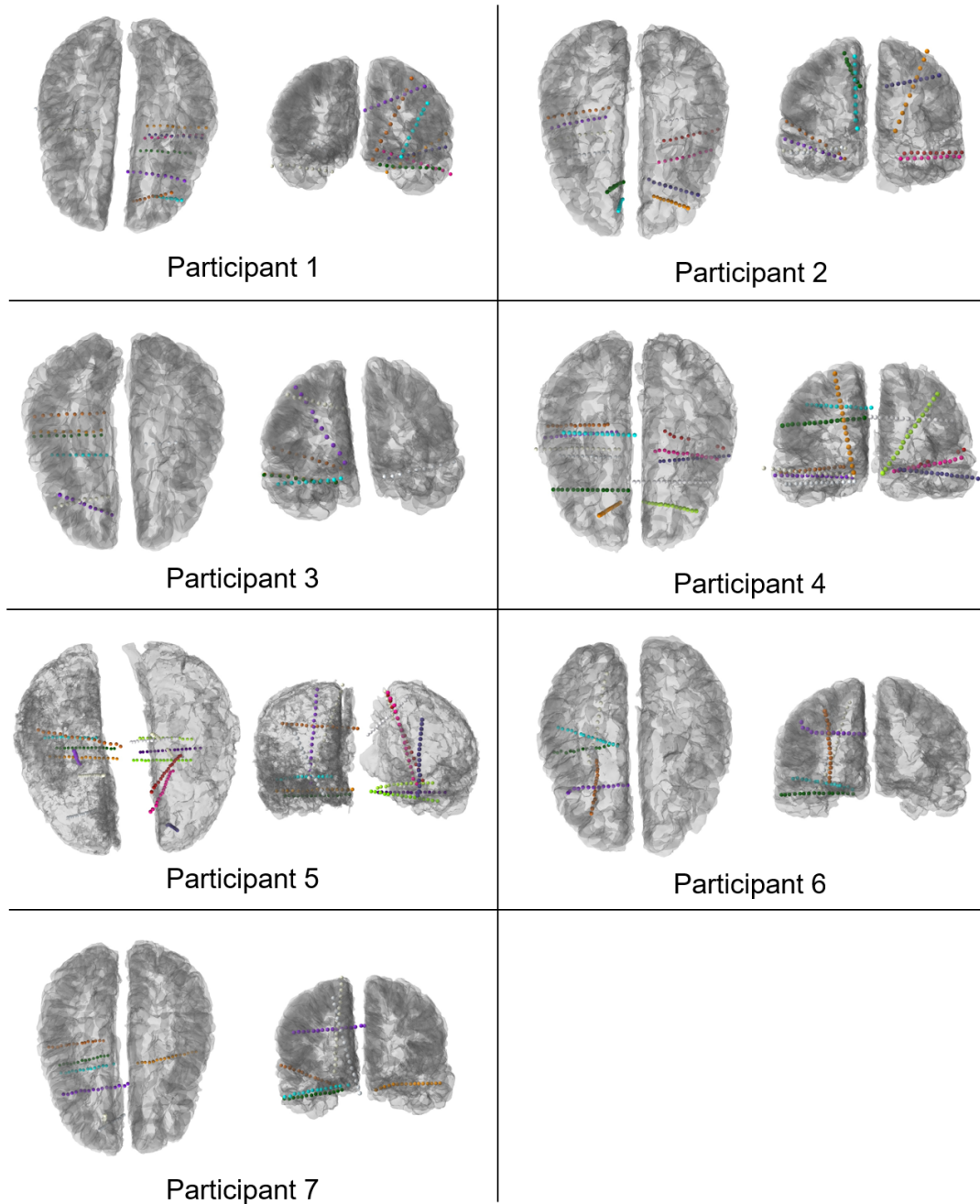


FIGURE 3.4: Electrode locations for the Harvard Sentences experiment.

"I was born not knowing, and have had only a little time to change that here and there."

Richard Feynman

4

Comparison of Data-Driven and Preconceived Model Features for Speech Activity Detection

4.1 Introduction

Traditional modeling approaches for BCI have commonly followed a framework similar to that of the diagram shown in Figure 4.1. Broadly, after the data acquisition stage of the process, the signal undergoes a set of signal processing

steps. Commonly, this includes pre-processing steps, whose goal is to attempt to mitigate confounding signal artifacts such as those from movement or faulty electrode connection. Following this, features of interest are extracted in some task-relevant manner from the conditioned signal for further use in a classification or regression machine learning model. Feature extraction, even in analyses that use deep learning methods, often still relies on static, researcher-engineered features [7, 110].

However, there are no standards with respect to the signal processing steps of this framework, and while there is overlap in themes, the specifics of signal processing pipelines are often unique to each research group. This paradigm reduces the reproducibility of BCI analyses and the robustness of BCI models. Evidence demonstrating that customary signal conditioning approaches are not necessarily required, or optimal, is valuable.

It is shown that, when using a deep learning approach, it is possible to exceed the performance of several established preprocessing methods on a speech activity detection task, while using only the raw channel data with no preprocessing or feature extraction from the signal. Reducing reliance on static signal processing methods removes significant barriers to reproducing results, and reduces implementation time and complexity for groups seeking to apply or test such methods on their own data.

Two signal processing methods from prominent published studies [56, 110] are emulated, both containing pre-processing and feature extraction steps. They are compared to a minimal signal processing paradigm, which contains no explicit feature extraction. The three methods are compared on their performance on a speech activity detection task.

Despite signal pre-processing in the spatial and time domains, and research-supported feature extraction, the two methods do not achieve the same level of

performance as the unprocessed approach. This shows evidence of the potential lack of generalizability or flexibility of preconceived feature sets over data-driven feature extraction methods.

4.2 Background

As previously mentioned of deep learning in the background in Section 2.3, it is a methodology where feature extraction is driven by the data, and in supervised learning cases, formulated as a classification or regression. Because it is an empirical method of feature extraction, it does not require assumptions of the underlying data. For example, traditional Principle Component Analysis, a common feature extraction method, cannot account for nonlinear relationships between input variables [163]. In contrast, an artificial neural network can theoretically represent any function [48]. The goal then becomes ensuring that the deep learning model is exposed to task-relevant information, and that the model architecture is complex enough to successfully extract meaningful features from the information.

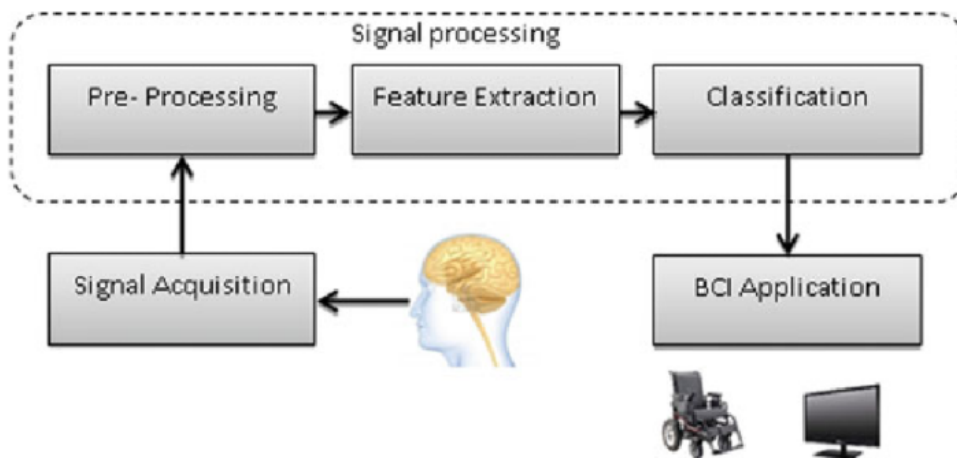


FIGURE 4.1: Diagram of traditional BCI modeling framework.

4.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) [90] are networks designed for learning filters which serve to reduce the dimensionality of the input signal by representing local features and connectivity, with deeper layers learning increasingly abstract feature representations. Early CNNs and precursors [45] were inspired by the hierarchical nature of simple and complex representations in the visual cortex [70], with a 1D version used for phoneme classification in 1989 [158]. In 2012, AlexNet [84], a CNN architecture, won the first ImageNet image classification challenge by a wide margin. It is often considered a genesis of the resurgence of interest in deep learning techniques, which has carried the field to where it is today.

A CNN architecture consists of CNN layers. A CNN layer learns a set of convolutional filters of arbitrary dimension, which process across the preceding network layer to create the following layer. The learned weights of each filter are applied to the entire input. The filters can be thought of as a method of parameter sharing, which greatly reduces the parameters in a CNN in contrast to

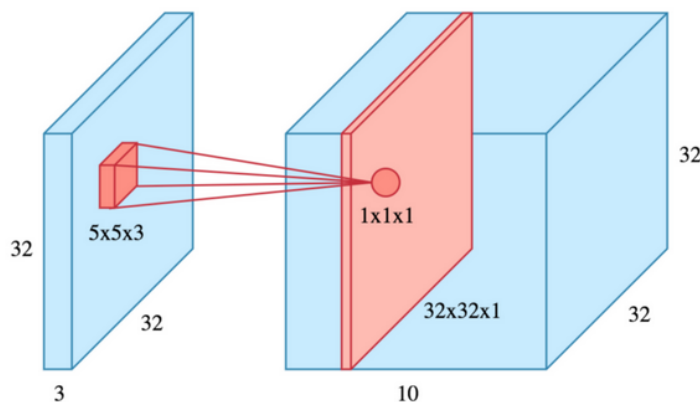


FIGURE 4.2: Diagram of CNN convolutional layer. Adapted from [36]

a fully connected network. Other layers can include pooling layers for further reducing the number of parameters. Figure 4.2 shows a diagram of the convolutional layer. The convolution filter calculates the element-wise multiplication and then summation of all elements, to produce a corresponding output in the feature map, and then processes a number of elements equal to the stride. Parameters that define a convolutional layer are the filter dimensions, the stride, the dilation, the zero padding of the input borders, and the number of filters to be learned. The initial convolution layer is applied to the input, creating a set of feature maps. Subsequent convolution layers apply the same process, but to the feature maps from the previous layer.

4.2.2 Prior Work

In 2019, Herff et. al. showed that it was possible to synthesize speech from ECoG signals using a non-deep-learning approach [56]. Another study by Angrick et. al. uses essentially the same signal processing as Herff et. al. [7], but instead synthesises speech using deep learning method: densely connected 3D CNNs. For the 3D DenseNet study, the extracted features are passed to a 3D CNN and regressed against the log Mel spectrogram at the corresponding time window. The reconstructed Mel spectrogram is then passed to an existing Wavenet Vocoder model, which creates an audio waveform from a spectrogram input. The method beats chance level on an objective intelligibility score. The signal processing method is chosen for this comparison because (1) the techniques used in preprocessing are common to the speech decoding domain, (2) the feature extraction paradigm is based on prior study findings of important frequencies and time scales, (3) the dataset for the study was collected using a similar experiment protocol to the Single Word dataset, and (4) the features were

successfully used in a study using a deep learning model for a speech synthesis task. Thus, this signal processing method provides a fitting basis for comparison to a CNN with no signal processing.

A landmark study by Moses et. al. used ECoG signals to perform online, real-time sentence classification from a patient with severe paralysis. The approach uses a combination of deep learning models to detect speech and classify words, followed by language models to produce sentences [110]. For the speech activity detection model, the study uses Long Short-Term Memory layers (LSTMs), which are a kind of Recurrent Neural Network, mapping sequences to sequences. In this case, the input sequence consists of the ECoG-derived features, and the output sequence is the probability of speech. The results yielded a 96% accuracy on speech activity detection. This study's signal processing approach is chosen because the pre-processing and feature extraction steps are used in several other breakthrough studies achieving state of the art results [8, 27, 100]. Additionally, the study also used a deep learning approach for speech activity detection. While this is a case study with only one participant, which would allow for significant model optimization, it provides an important datum for comparison.

4.3 Methods and Model

4.3.1 Dataset

The Single Word dataset, detailed in Section 3.1, is used for the comparison of three preprocessing and feature extraction methods on the task of speech activity detection. Briefly summarizing, during the Single Word experiment, 5 participants were presented and then read aloud a set of words while ECoG signals

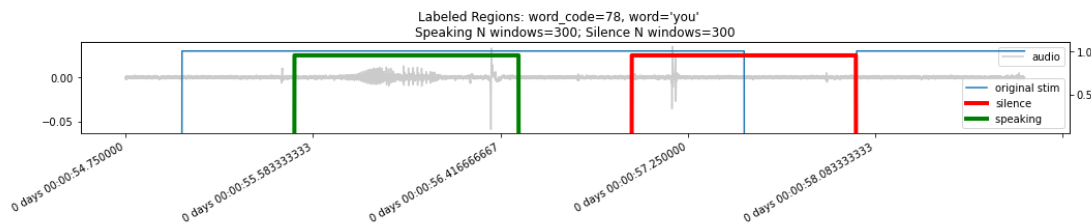


FIGURE 4.3: Labeling Scheme for Speech versus Non-speech for the Single Word experiment data.

and voice audio were being recorded.

Data Labeling

Speech labels used for training the models on the task of speech activity detection are created in reference to the stimulus cue of the word being presented in the experiment. Every time-sample from 0.5 seconds after the word presentation cue to 1.5 seconds after the cue are labeled as ‘speaking’. Every time-sample from 2.0 seconds after the word presentation cue to 3.0 seconds after the cue are labeled as ‘not speaking’. The other segments, from the cue to 0.5 seconds after, and from 1.5 to 2.0 seconds after, are purposefully left unlabeled. Figure 4.3 shows an example of the labeling scheme.

This labeling scheme is chosen based on the stimulus presentation cue, as opposed to direct energy detection in the audio signal, in order to develop a more robust model that does not directly rely upon the acoustic signal. This is done to emulate the scenario where the user is unable to speak, thus precise labels for the presence or absence of speech would not be available. Instead, the proposed labeling indicates the time segments where speech is most expected, which can be generalized to imagined speech.

4.3.2 Signal Processing Protocols

The details of the three signal processing methods being compared are provided below. The two replicated methods are presented as they are implemented on the Single Word dataset rather than details from their source study. Specifically, the sampling rate of the Single Word data differs from that of the data used in the other studies.

The Single Word ECoG signals as well as the accompanying ‘speaking’ and ‘not-speaking’ label targets, are both sampled at 1200 Hz. All signal processing procedures including filters are implemented using Matlab ver. 2017b [104].

Signal Processing Method from Herff et. al.

Figure 4.4 shows a diagram of the method. The following steps are performed for each ECoG electrode channel of the participant, across time. First, the signal is downsampled from 1200 Hz to 600 Hz. The data are then linearly de-trended, where a least squares regression line is fit to the data and then subtracted. Next, the data are filtered to the broadband gamma frequency range, 70-170 Hz, using an IIR pass-band filter, followed by an IIR notch filter from 118-122 Hz to remove the 60 Hz harmonic. The logarithm of the power of the resulting broadband gamma signal is calculated.

For each channel, the broadband gamma is processed in 450 ms sliding windows. The signals in the window are decimated to 20 Hz, resulting in one average log band power measure for each 50 ms. The 9 time samples across the 450 ms window are flattened into a feature vector containing high gamma information over an approximately 0.5 s period for all electrodes. The feature vector is then normalized to unit mean and variance. The window is advanced by 10 ms,

where another feature vector is created. This results in a final signal of dimension $\#channels \times 9$ sampled at 100 Hz.

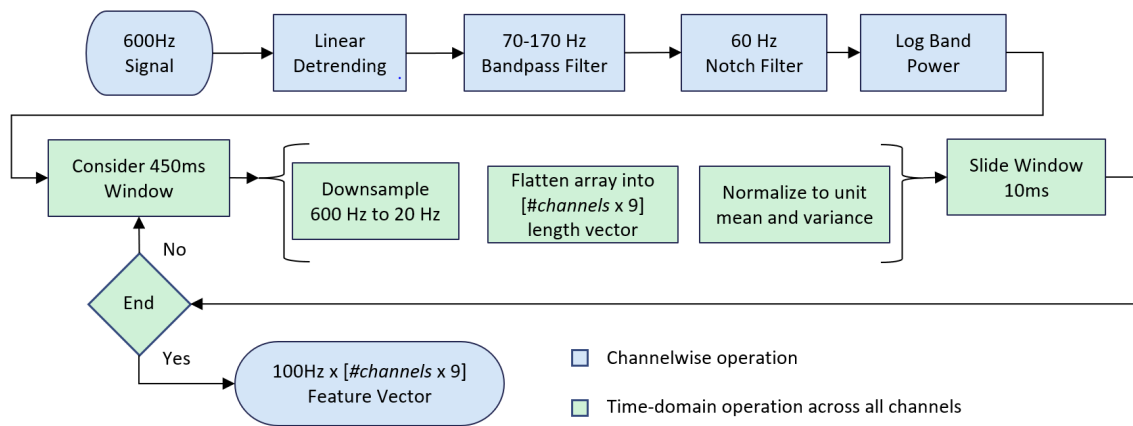


FIGURE 4.4: Diagram of the Herff signal processing method.

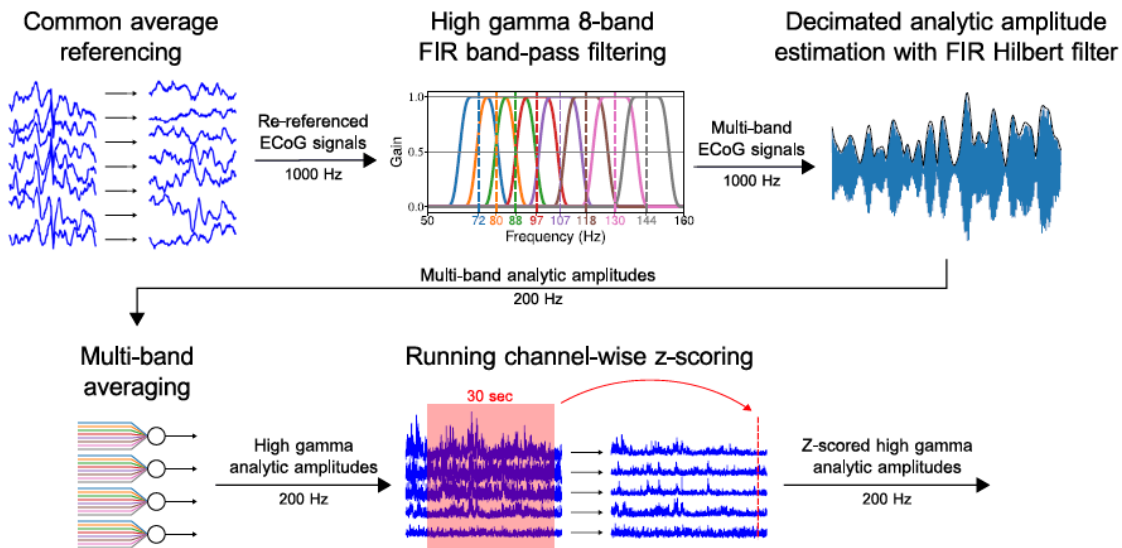


FIGURE 4.5: Diagram of the Moses signal processing method. Adapted from [110]

Signal Processing Method from Moses et. al.

The original study used an input signal sampling rate of 1000 Hz. In this implementation, the native dataset sampling rate of 1200 Hz is maintained until downsampling to 200 Hz. Figure 4.5 shows a diagram of the process. First, a common average reference is applied to all ECoG channels. Then, a set of eight FIR filters are created with the following pass-bands: 73.0 ± 4.68 , 79.5 ± 4.92 , 87.8 ± 5.17 , 96.9 ± 5.43 , 107.0 ± 5.70 , 118.1 ± 5.99 , 130.4 ± 6.30 , 144.0 ± 6.62 . The ECoG signals are filtered through each pass-band filter, resulting in 8 band-passed signals per electrode channel. The amplitude envelope of all signals is calculated using the Hilbert transform. The signals are decimated from 1200 Hz to 200 Hz. Then, for each channel, the 8 band-passed amplitude envelope signals are averaged together. Finally, the signals are z-scored within a 30-second sliding window.

Minimal Signal Processing Method

In order to compare to the other signal processing methods with no feature extraction, the Base approach simply downsamples the ECoG signal from 1200 Hz to 600 Hz. This is done in order to directly compare the other two signal processing methods in terms of sampling rate, which is selected due to the relevant bandwidth of broadband gamma features. Additionally, this has the benefit of reducing the associated model size. With a sampling rate of 600 Hz, information of frequencies up to 300 Hz is maintained, and speech processes are thought to appear generally in the high-gamma band with little information occurring in frequencies above 250 Hz [19]. Other than downsampling the signal, no other conditioning is done prior to model input for classification.

4.3.3 Method of Comparison

The three signal processing approaches are applied to the same set of data from the Single Word task. After processing, this yields a signal at $200\text{Hz} \times \text{channels}$ for the Moses approach, $100\text{Hz} \times (5 \times \text{channels})$ for the Herff approach, and $600\text{Hz} \times \text{channels}$ for the baseline approach. The signals are trained on a speech activity detection task using a CNN model. While the differences in input signal preclude the CNN architecture from being identical for all three methods, the differences in models are minimal to ensure an analogous, if not equivalent, architecture for a more direct comparison of results. Figure 4.6 is a summary overview of the comparison. The architecture of the CNN models is covered in detail below. Broadly, the model operates on a 0.5 s window of signal, for all features or time samples, and the target for each respective window corresponds to whether the participant is ‘speaking’ or ‘not-speaking’ at the end of the 0.5 s window. This problem formulation is intended to explore groundwork for the design of a causal speech activity detection model, which is presented in Chapter 5.

The metric for comparison is the conventional accuracy of the model, as ‘speaking’ and ‘not-speaking’ window trials are balanced. For each participant and method, the training and evaluation process is repeated 3 times. Further, a CNN model variant with a greater number of CNN filters is trained in order to verify that model size is not a determining factor in performance.

4.3.4 CNN Models

The CNN architecture implemented for the comparison is informed by previous work applying CNNs to non-invasive EEG neural data [89, 140], and by model optimization to the current task.

The architecture shown in Figure 4.7 is the CNN for the Base model corresponding to the unprocessed signal processing scheme. The input to the CNN is a timeseries array that is 0.5 s of time samples, equating to 300 samples for the 600 Hz signal. The first convolutional layers aggregate across time with kernels and stride of five samples, and a dilation of two samples to further downsample. The next layer maintains the kernels size and stride, but returns to a default dilation of one. The remaining two convolutional layers learn 3x3 kernels with unit stride and dilation until a final dense layer outputs to a sigmoid activation. A total of 16 filters are learned in each convolutional layer. The default number of CNN filters learned at each layer is 32.

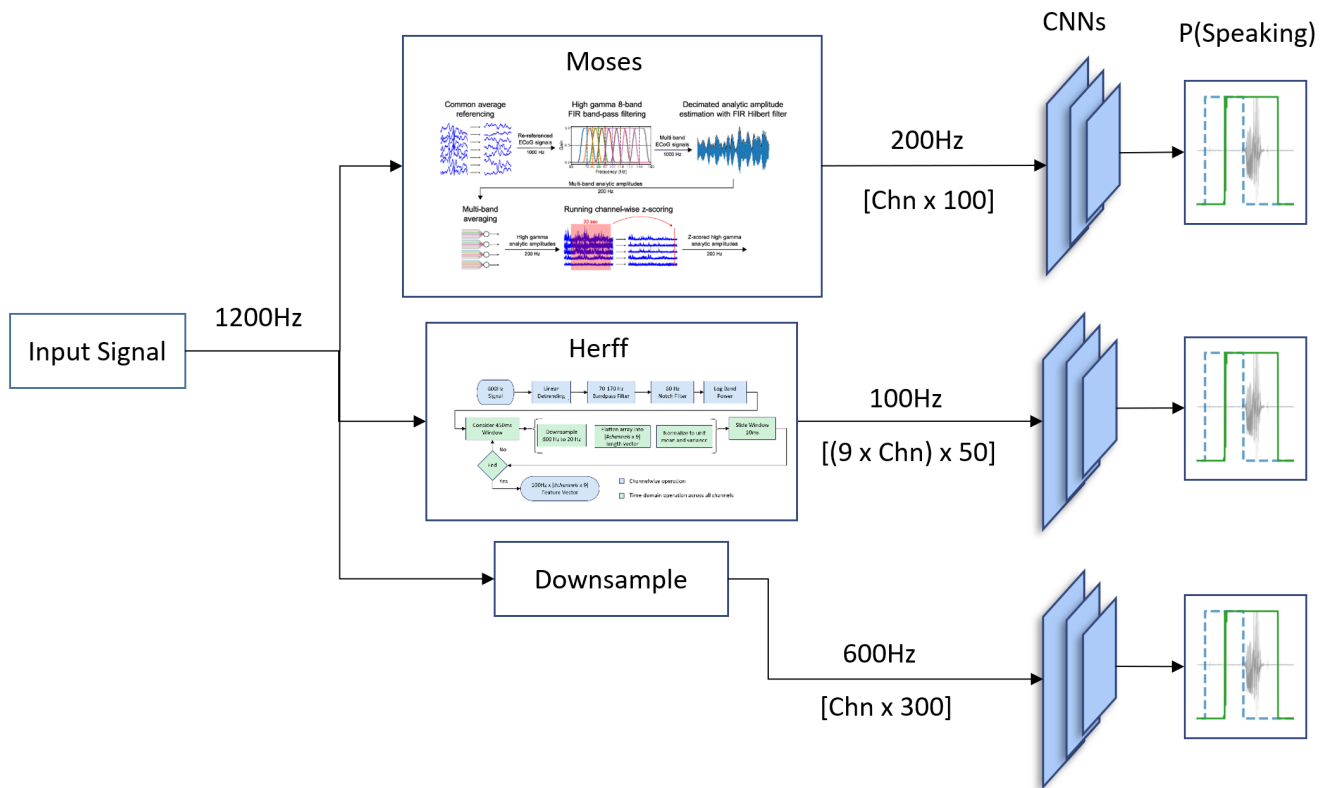


FIGURE 4.6: Diagram of the three signal processing methods. The output of the CNN returns a probability of 'speaking' or 'not-speaking' at every time sample.

In order to keep the model architectures as similar as possible for direct comparison, while still accommodating for the differences in input shape, small changes are made to the architectures used for the Moses and Herff models. Both architectures maintain the same number of layers; however, the filter size and stride of the first two 1D convolutional layers are changed in order for the 3x3 filter layers to act upon approximately similar dimensions. For the Moses process, with a 200 Hz sample rate, the first two layers have a kernel size and stride of 3.

Optimizer. The CNNs use stochastic gradient descent from gradients produced by error back-propagation. The Adam optimizer proposed in [80] is used in this work the learning rate fixed to $\alpha = 0.001$ for all experiments and models.

Loss Criteria. Binary cross-entropy loss is used as the model's objective criteria. Models are evaluated through multiple refits using a K-Fold procedure

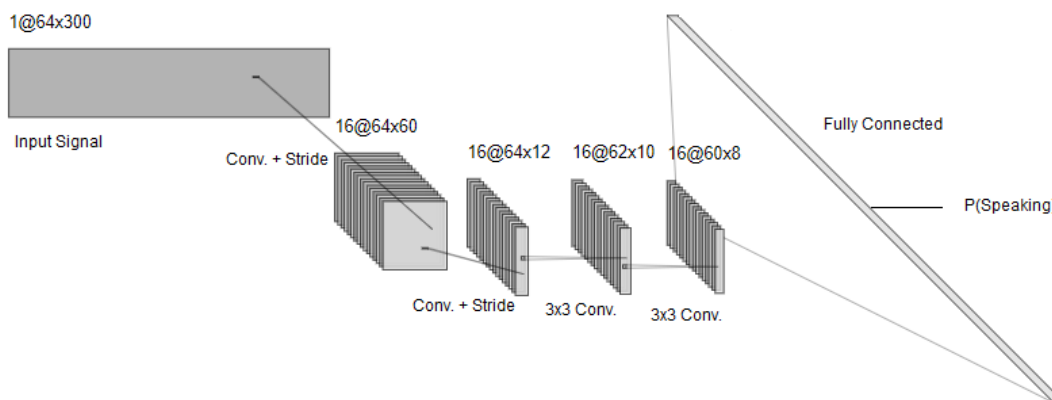


FIGURE 4.7: Base CNN model architecture. The first two layers apply temporal convolution with a filter length and stride 5 for downsampling. The following two convolution layers learn 3x3 filters. All convolution layers use a Rectified Linear Unit (ReLU) for their activation functions. Following this is the fully connected layer. The activation function of the final layer is a sigmoid, which outputs a binary classification, the probability the input sample is 'speaking'.

across participants' trial sessions. A single holdout trial is used for evaluation in each fold and the remaining folds are used for training. For our experiments, the best model on the cross-validation is maintained and stored after 100 epochs of training.

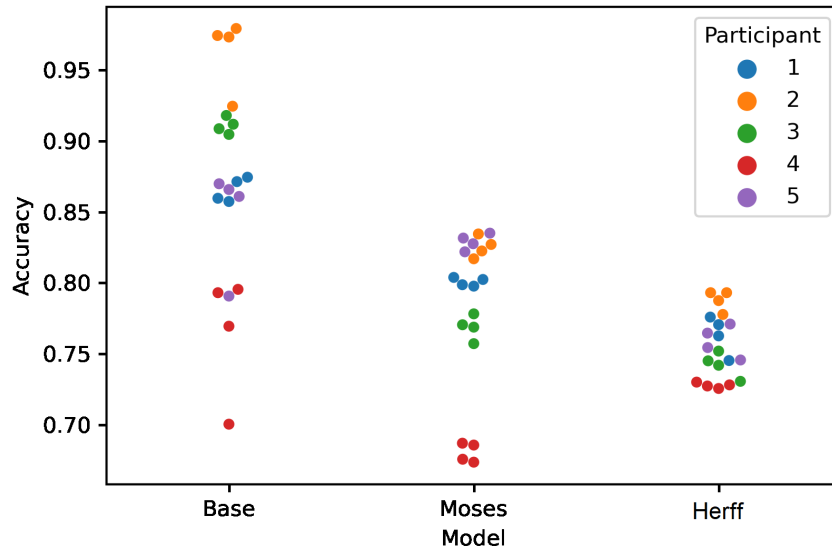
4.4 Results

Accuracies of each model and participant, as well as the average across participants, are reported in Table 4.1. The percentages in the table represent the average of the 4 train-and-test repetitions for each participant and model. Differences in the three signal processing methods are evident, with the unprocessed method outperforming the other two methods at an average accuracy of 86.5%. The Moses method performs slightly better than the Herff method, with 78.1% and 75.4% respectively.

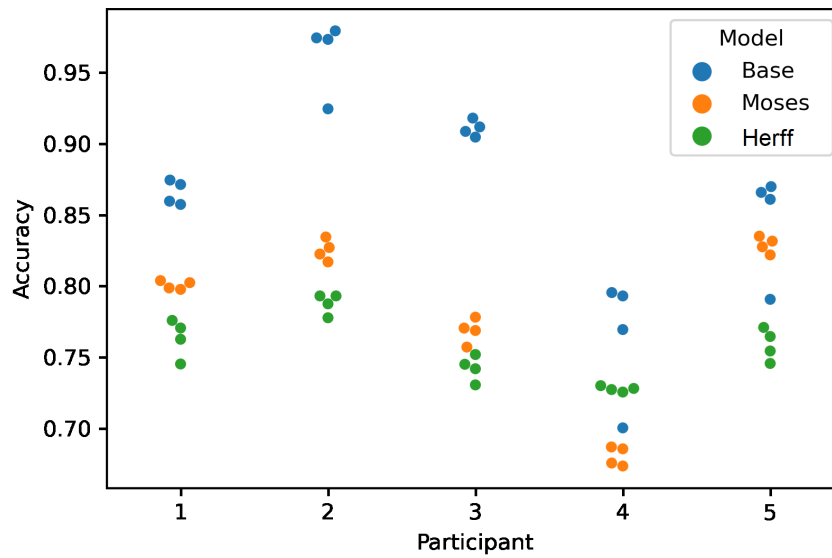
Participant	Base	Moses	Herff
1	86.6%	80.1%	76.2%
2	94.1%	82.5%	79.0%
3	91.2%	76.9%	75.9%
4	74.4%	68.1%	73.0%
5	82.7%	82.9%	72.7%
Average	86.5%	78.1%	75.4%

TABLE 4.1: Model accuracies by participant for comparison of signal processing methods.

In general, participants perform similarly well in comparison to other participants across all three tasks. Participant 2 performs best overall, with the highest accuracy on the Base and Herff models, and second highest on the Moses model. Participant 4 has the worst performance across models, with the lowest accuracy on the Base and Moses models, and the second lowest on the Herff model.



(A) Accuracy by model, separated by color for each participant.



(B) Accuracy by participant, separated by color for each model.

FIGURE 4.8

Figures 4.8a and 4.8b show two swarm plots of accuracies for all model/participant combinations. The top figure is grouped by model, stratified by color for each

participant. The bottom figure reverses the paradigm, being grouped by participant, and stratified by color for each model. Figure 4.8a shows that the Base model, while having the best overall accuracy, has a wider spread than the processed models. In particular, the Herff method appears significantly more participant-invariant than the other two methods, which have clear clusters for each participant.

Model Complexity Analysis

In order to verify that the superior performance of the unprocessed approach is not simply driven by model complexity, several variations of the Moses and Herff models are tested for comparison. First, the number of CNN filters is increased from 32 to 48, concurrently increasing the number of model parameters. Second, alternative kernel size and stride configurations are tested for the first two 1D-CNN layers of the models. The best performing such configuration, other than the default described in Section 4.3.4, is a kernel size of 10 with a stride of 3 for the Moses model, and a stride of 2 for the Herff method. Table 4.2 reports the average accuracy across all participants, as well as the average parameters and number of CNN filters learned, for each model configuration in comparison to the original three.

	Avg. Accuracy	Avg. Parameters	# CNN Filters
Base	86.5%	38,305	32
Moses	78.1%	36,218	32
Herff	75.4%	116,421	32
Moses	78.2%	68,770	48
Herff	77.6%	185,186	48
Moses 10-kernel	80.5%	34,352	32
Herff 10-kernel	73.3%	137,541	32

TABLE 4.2: CNN model complexity adjudication

4.5 Discussion

The proposed unprocessed speech activity detection approach, on average, exceeds the performance of the two comparison signal processing methods. The unprocessed approach has the largest variance of the three methods, as shown in Figure 4.8a. The excellent performance of Participant 2 increases the average; however, the poor performance of Participant 4 decreases it considerably. If Participant 4 were to be excluded, with the exception of one trial for Participant 5, all unprocessed trials would exceed the best-performing trial from either of the other methods.

For the three methods, there is a correlation between variance and sampling frequency. That is, the method with the largest frequency, 300 Hz for the unprocessed method, has the largest variance in trial performance, while the method with the smallest frequency, 100 Hz for the Herff method, has the smallest variance in trial performance. It is unclear if this relationship is causal. Empirically, it is observed that the Herff method, has the most participant-independent performance of the three methods.

Moses et. al. [110] and Makin et. al. [100] both used the Moses method, and applied speech activity detection as part of a larger speech decoding analysis, though using RNN and LSTM models rather than CNNs. Although they did not report specific results on speech detection performance, it is implied that the performance exceeded the levels achieved in this analysis. However, without a direct comparison to an unprocessed approach, it remains unknown whether the same or better results could have been achieved without the feature extraction process.

This analysis intends not to assert that automatic, data-driven, feature extraction is superior to researcher-engineered feature extraction in every case.

Rather, it is to show evidence to support rejecting the null hypothesis that the static features will always yield a better result. The results imply that, when using a method such as deep learning that includes automatic feature extraction, the practice of preconceived feature extraction prior to model input could either be redundant, or perhaps filter out information that is potentially useful to the classification task. This is explored further in the following chapter.

*"It is good to have an end to journey toward; but it is
the journey that matters, in the end."*

Ursula K. Le Guin

5

SincIEEG: An Interpretable Deep Learning Model for Speech Activity Detection using ECoG Signals

5.1 Introduction

Neural speech decoding systems have made significant progress, including describing brain regions and mechanisms involved in speech, predicting words or phonemes, and translating neural signals to articulatory kinematics, text, or

directly to speech waveforms [8, 19, 26, 27, 55, 98, 112]. Recent efforts have progressed to real-time decoding and synthesis of overt and imagined speech [6, 56, 101, 103, 110, 111]. While these studies primarily focus on broadband gamma activity ($\sim 70\text{-}250$ Hz), recent studies have shown that traditional lower-band frequencies ($\sim 0\text{-}50$ Hz) also contain relevant and complementary information for speech decoding [121].

Deep learning has been demonstrated to be an effective method for decoding speech from ECoG signals and its inclusion in the decoding and synthesis pipeline has increased in recent years [7, 8, 100, 110]. Although an end-to-end architecture may eventually be wholly effective with sufficient training data, some current approaches have adopted a modular scheme with several sequential component models, each configured for a specific aspect of the speech decoding process [78, 110, 111].

Regardless of the specific approach, the overarching goal is to decode imagined or attempted speech directly from brain signals to provide an alternate communication channel for those who have lost the ability to speak. Here, the goal is not to maximize a metric for the quality of speech decoding. Instead, the approach is conceived from the perspective of identifying brain activity associated with intervals of intended speech output, with the ultimate objective of reliably detecting activity associated with imagined speech.

The present work introduces a component model, SincIEEG, based on a convolutional neural network (CNN) architecture developed for the task of speech activity detection [128]. The model is designed as a gateway, constantly monitoring brain activity to identify the segments pertinent to speech production. These detected segments can then be sent to downstream models for subsequent speech decoding and synthesis. SincIEEG, unlike a traditional CNN, learns a

set of bandpass filter coefficients at its input layer. This provides several advantages over a traditional CNN, as the number of required model parameters is significantly reduced in comparison, making it computationally efficient in terms of implementation. This compactness allows for flexibility without increasing the optimization problem. Moreover, unlike most traditional CNNs, the SincIEEG model has the distinct advantage of yielding interpretable parameters. The bandpass filters learned by SincIEEG can be visualized and equated to conventional spectral brain features.

The results demonstrate that SincIEEG is capable of detecting the presence or absence of speech during each time interval with a high level of accuracy, and compare the model's performance to a traditional CNN model, as well as non-deep learning methods. In addition, the generalizability of the model architecture is highlighted in terms of providing empirical, interpretable insights about the discriminable bandpass spectral features for any physiological data that can be represented as an aggregate of bandpass activity.

5.2 Materials and Methods

5.2.1 Dataset

The dataset used for the development and evaluation of the SincIEEG model architecture is the Single Word ECOG dataset presented in Section 3.1. In order to maintain continuity and reduce confounding variables, the dataset, preprocessing, and labeling, for the speech activity detection experiment were maintained from Chapter 4. Briefly, the Single Word dataset has 5 participants cued to read single words aloud while both ECOG neural signals and audio was synchronously recorded. For the speech activity detection, 0.5 s of ECOG data were

considered and labeling was a simple scheme based on timing of experiment cues.

5.3 Model Design and Optimization

The SincIEEG model is a Multi-SincNet based convolutional deep learning architecture adapted for real-time detection of human speech from ECoG input signals. Proposed in [147] for hand-pose classification from myoelectric sensor readings, and based on the work from [128], the Multi-SincNet architecture learns the coefficients of a set of parallel finite impulse response (FIR) bandpass filters, applied across the input channels. Subsequent convolutional layers learn kernels that aggregate across time and bandpass frequency dimensions. A final global view, established by a fully connected layer and sigmoid activation,

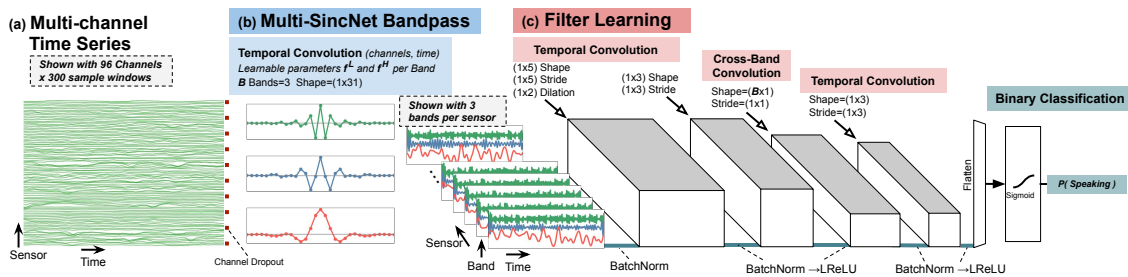


FIGURE 5.1: The **SincIEEG** deep learning architecture: a classification model composed of a Multi-SincNet input layer and multiple subsequent convolutional layers. **(a)** SincIEEG takes raw multi-channel ECoG time series data as input, with channel dropout for improved regularization. **(b)** Multi-SincNet learns bandpass filter parameters to decompose the input signal - illustrated here with three pass-bands. **(c)** The filtered signals are normalized with respect to the band dimension using spatial normalization before convolutional layers learn kernels across time and pass-bands. All hidden layers use batch normalization for regularization and *Leaky Rectified Linear Units* for activation. The model predicts the likelihood of speaking using a *Sigmoid* activation at its output layer.

classifies either ‘speaking’ or ‘not-speaking’ from labeled data. Figure 5.1 illustrates the SincIEEG model and its layer configurations. This section details the architecture and training strategy to produce models for validation described in Section 5.4.

In overview, the inputs to the model are 500 ms windows of raw IEEG data (300 time samples) with a stride of 2 ms (1 time sample). Each 500 ms window represents one training sample for the model, described in Section 4.3.1. A model was trained for each participant, using all of the quality electrodes available. Electrodes over the auditory cortex were excluded for a model validation check, detailed in Section 5.4.3. A K-fold training methodology was used and is detailed further in Section 5.3.5.

This architecture was developed and implemented using Pytorch [117] deep learning Python library. Other critical software libraries used for development and discovery include matplotlib [72], numpy [50], pandas [150] [106], seaborn [159], and SciPy [155].

5.3.1 Multi-SincNet Input Convolution

The first layer in the SincIEEG model is a Multi-SincNet layer, an extension to the the Kaldi speech framework’s [129] SincNet, which applies a SincNet to each of the incoming sensor channels. A SincNet layer learns a configurable number of bandpass filters, parameterized through two cutoff frequencies, f_L and f_H . The Multi-SincNet layer can therefore be used to decompose a collection input signals into a fixed set of learned bands.

In equations 5.3.1 and 5.3.1, multiple filters are conceptualized as vectors of low and high cutoffs, F_L and F_H respectively, identifying regions of the input’s

spectrum that the model uses for classification. These vectors are a parameterization of a SincNet layer, which is shared in the experiments across all sensors $s \in S$.

$$F_L = \{f_L^0, f_L^1, \dots, f_L^{i=B-1}\} \in \mathbb{R}^+$$

$$F_H = \{f_H^0, f_H^1, \dots, f_H^{i=B-1}\} \in \mathbb{R}^+$$

$$K : (f_L, f_H, f_s) \mapsto \mathbb{R}^W$$

$$\text{SincNet}(F_L, F_H) = \{K(F_L(i), F_H(i))\}$$

$$\text{Multi-SincNet} = \text{SincNet}_{F_L, F_H}(s) \quad s \in S$$

Sharing bandpass filters across each sensor reduces parameters, improves model latency, and regularizes the treatment of sensor data.

Each FIR filter, k is implemented as a set of kernel coefficients and applied through convolution with the input signal X .

$$X \otimes k_{(f_L, f_H)} = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} X[i] * k_{(f_L, f_H)}[j - i]$$

Where X is the input signal and k_{f_L, f_H} is the vector of kernel coefficients that allows frequencies in $[f_L, f_H]$ to remain in the signal. Additional details on the calculation of k coefficients and how they compare to learned kernels can be found in [128].

Filters are initialized to uniformly sub-divide the majority of the available spectrum (i.e., 0-300 Hz) with a 3 Hz region of overlap between adjacent bands. The original Kaldi implementation initializes bands starting at a low-cutoff of 30 Hz, but this minimum starting frequency is reduced to 10 Hz for the present analysis to help encourage the use of lower frequencies that may be relevant for this application [121]. The Kaldi SincNet implementation also includes a minimum frequency and minimum bandwidth constraint, which are configured to be 1 Hz and 3 Hz, respectively. Kaldi enforces these minimums by increasing

the absolute value of the learned low-cutoffs and bandwidths by their respective minimums. Future work should explore the impact of different potential initialization schemes.

5.3.2 Activation

Rectified linear units (ReLU), defined as $y = \max(0, x)$, provide a linear gradient for all input $x \in \mathbb{R}^+$ and 0 gradient for $x \leq 0$. With zero-centered bandpass outputs, a large portion of values will not have a gradient with ReLU activation. Instead, the Leaky ReLU activation (LReLU) provides a small gradient for $x \leq 0$, while still being non-linear and computationally simple. The LReLU activation is defined in equation 5.3.2, where the default $\alpha = 0.01$ is used for all experiments.

$$\text{Leaky ReLU}(x) = \max(0, x) + \alpha * \min(0, x)$$

Using LReLU on zero-centered data still greatly diminishes negative inputs. However, the learned affine parameters within the batch normalization layers can learn to offset any inputs into regions with higher variance.

5.3.3 Batch Normalization

The amplitude of the output from the Multi-SincNet filters scale directly with the amplitude of the input signal. Between-sensor relative magnitudes are important to maintain, so scaling at the sensor dimension of intermediate data is avoided in the early layers.

Brain dynamics are not evenly distributed in the frequency domain, however, and will tend to have higher amplitudes at lower frequencies. This means

the additional bandpass dimensions may be distributed at different scales, making it difficult to learn shared kernels in subsequent convolution layers. Furthermore, the scale of the intermediate values may shift as the cutoff frequencies of the learned bandpass filters are optimized.

Therefore, in order to balance influence when learning kernels applied across bands, and to scale hidden outputs to activation regions, a spatial batch normalization [74] is applied at the band dimension in the three hidden outputs following the Multi-SincNet input layer. Re-scaling each band independently maintains within-band relative dynamics that can be learned using shared weights.

$$\begin{aligned}\mu_f &= \frac{1}{BST} \sum_{b=0}^{B-1} \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} X[b, s, f, t] \\ \sigma_f &= \frac{1}{BST} \sum_{b=0}^{B-1} \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} (X[b, s, f, t] - \mu_f)^2 \\ y &= X - \mu_f \frac{\sqrt{\sigma_f + \epsilon * \gamma + \beta}}{\sigma_f} \\ &\text{for } f \in F\end{aligned}$$

Where B is the batch size, S is the set of sensors, F is the set of bandpass regions, and T is the number of input samples. Learned affine parameters β and γ allow the model to adjust the center and scale away from the origin and unit variance. Following cross-band convolution, spatial normalization is applied across sensors - computing μ_s and σ_s analogous to μ_f and σ_f . At this point in the architecture, distributions across sensors are well-normalized and suitable for batch normalization's regularizing effect, reducing internal covariate drift.

5.3.4 Monte Carlo Dropout

Sensor systems with many highly responsive input channels may have spurious errors or drift, and sometimes must be removed in pre-processing. Additionally, for general tasks such as speech activity detection from an ECoG array, some important brain regions may have multiple sensors covering them, resulting in

high co-linearity across channels. To regularize co-linearity across sensors, channel dropout [151] is applied on the input to the model during training. Channel dropout on the sensors zeros all signal values for a sensor with an independent Bernoulli random number parameterized by probability p . It is common to avoid using dropout when using batch normalization since the noise caused by the dropout will skew the mean and variance statistics used in normalization towards zero. However, for SincIEEG, the data modality is already centered at zero, and the practical application motivates robustness to sensor dropout.

5.3.5 Optimization Procedure

All deep learning models in this work, both the SincIEEG described above and CNN model described in Section 5.4.3, use stochastic gradient descent from gradients produced by error back-propagation. The Adam optimizer [80] is employed with the learning rate fixed to $\alpha = 0.001$ for all experiments. Binary cross-entropy loss between the target label and the model's output is used as the objective criteria.

Models are evaluated through multiple refits using a K-Fold procedure across a participant's sessions. A single holdout session is used for evaluation in each fold and the remaining sessions are used for training. Some participants had three sessions, providing two training sessions per fold, while others had only two sessions overall and provided one session per training fold. The training data is randomly split into a 25% cross-validation portion for monitoring model performance during training. After each epoch of training, a model under optimization is applied to the cross-validation data and scored. For the SincIEEG and CNN experiments, the best model on the cross-validation is maintained and stored after 100 epochs of training.

Experiments without auditory sensors and other supplementary architecture exploration used early stopping. For these experiments, if the cross-validation performance did not improve for 10 epochs during training, then the best model at that point was stored and the training procedure ended. The early stopping procedure generally produced models with similar performance to their 100 epoch counterparts. Other configurations that were explored using this truncated procedure include variations of activation function, batch normalization, number of learned kernels, and other modifications to convolution configuration. Performance was robust for most configurations and these preliminary experiments focused on reducing model complexity.

5.4 Model Validation

ECoG data acquired from participants performing the speech task were used to further validate the model. The models are validated both quantitatively for predictive performance, as well as qualitatively for convergence of the spectral band filters to physiologically plausible ranges.

5.4.1 Prediction Accuracy

The prediction accuracy is simply computed as the proportion of windows correctly classified as ‘speaking’ or ‘not-speaking’. Visualizations that overlay the stimulus cue, curated labels, speech audio signal, and the model’s predicted likelihood of speech are presented. Aligning recorded speech with model predictions across multiple training windows enables an examination of the model’s predictions with both the labeled regions and recorded speech data. The model’s ability to predict speech occurring outside the labeled region help to validate

the model's generalization capabilities. Ultimately, this visualization provides an indication as to how the model would perform in practice. For instance, frequent oscillations in the predicted likelihood may achieve reasonable accuracy but ultimately be unreliable for use in a classification pipeline.

5.4.2 Spectral Band Convergence

A key aspect of this model's utility is its ability to learn spectral bands that minimize the loss function of the network. When the band parameters are combined with the loss and cross-validation loss for each training batch, a visualization of the band convergence over time can be obtained. This visualization can serve several purposes. For the present analysis, it serves as an additional method of model vetting and interpretation, to establish the frequency bands the model identified as empirically predictive. For other analyses, it could serve as an exploratory tool to investigate whether frequency information is central to the phenomenon.

5.4.3 Comparison Models and Benchmarks

Randomization Tests

In order to compare the model performance to random chance, the model prediction was assessed when trained on randomly labeled segments. The labeling scheme maintained a proportional amount of speaking/not-speaking labels, and thus the chance accuracy should be 50%. To confirm this, the train and test paradigms were kept identical, except that before training, a labeled segment was randomly assigned a 'speaking' or 'not-speaking' label. The hyperparameters chosen for model configuration were 1-Band with a dropout of $P = 0.5$.

Auditory Cortex Electrode Removal

To verify that classification performance was not merely being driven by auditory feedback, electrodes in the auditory cortex region were identified based on anatomical landmarks and removed from the analysis (see Figure 7.2). An abbreviated evaluation of SincIEEG was performed to confirm that the classification performance was not significantly degraded by the exclusion of the auditory electrodes. Optimization time of these additional models was reduced by using early stopping as described in Section 5.3.5. Additional testing verified that early stopping does not unfavorably bias the resulting model performance.

LDA and SVM Benchmarks

To explore whether the frequency bands that the SincIEEG model identified would confer some benefit over using the entire broadband spectrum, the performance using the bands that 3-band SincIEEG learned for each participant was compared to the performance using broadband activity from 0.5-170 Hz frequencies. The 3-band version was chosen to compare because it is more distinct from broadband than the 5-band version which generally occupies a greater proportion of the spectrum. A Linear Discriminant Analysis (LDA) and a linear Support Vector Machine (SVM) were implemented as performance benchmarks. Because these comparatively simple classifiers are not capable of attaining reasonable performance using raw ECoG timesamples, a preprocessing method derived from [56] was implemented that generates a band power aggregate measure over a 500 ms window that updates every 50 ms. The labels were accordingly downsampled to 20 Hz. For each label, the preceding 500 ms of the corresponding preprocessed ECoG signals were used to compute the input features. The resulting feature array was flattened into a vector for training

the LDA and SVM models. This process was performed for both the broadband and 3-band SincIEEG versions.

Standard CNN

To establish how SincIEEG performs compared to a traditional deep learning method, a standard CNN was implemented and evaluated based on [140]. For this CNN, the first convolutional layers aggregate across time with kernels and stride of five samples, and a dilation of two samples to further downsample. The next layer maintains the kernel's size and stride, but returns to the default dilation of one. The remaining two convolutional layers learn 3x3 kernels with unit stride and dilation until a final dense layer outputs to a sigmoid activation. A total of 16 filters were learned in each convolutional layer. The standard convolutional network model is an important alternative to SincIEEG as it uses the same convolution operation but is not directly interpretable. The training and testing paradigms remained unchanged, only the model architecture was exchanged.

5.5 Results

5.5.1 Prediction Accuracy

The average SincIEEG model accuracy across all participants was 94.1% (s.e. 3.5%), and all but one participant achieved an accuracy above 90%. Figure 5.2 shows the accuracy of all model configurations per participant with each configuration repeated three times. Results from Participants 1 and 2 were very consistent regardless of hyperparameter, while Participant 3 showed significant variability in the 3- and 5- band versions, and Participant 5 performed better

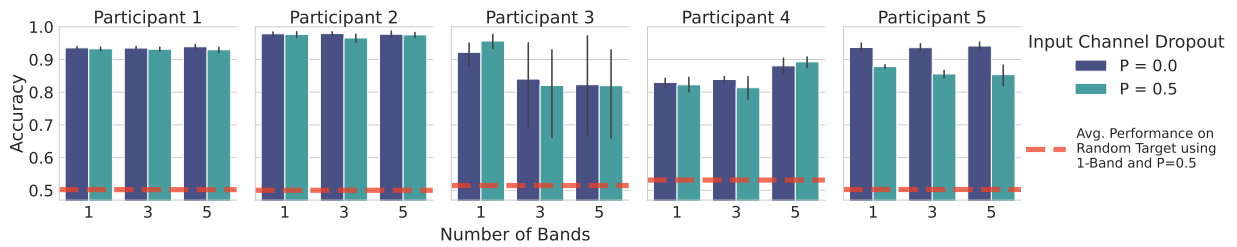


FIGURE 5.2: Mean and variance of accuracy for all repetitions' test folds, for each participant model configuration.

without dropout. These differences are most likely mediated by electrode number and placement. However, the ability of the model to achieve good performance on such a variety of electrode locations is a testament to its robustness, and the advantages of a participant-specific feature set.

As described in Section 4.3.1, target labels were created from the timings of experiment cues, rather than the participant's speech. Therefore, to better gauge speech detection performance for practical speech detection applications, predictions were qualitatively assessed by visual inspection into one of three categories: *Full Success*, *Partial Success*, and *Failure*.

A word trial was considered a *Full Success* if the prediction captured the entirety of the spoken word prior to onset and maintained until speech had ceased. Subplots (a), (d), and (g) in Figure 5.3 are examples of *Full Success* trials. Regions of false positive predictions encompassing a correctly identified speaking region were still categorized as a *Full Success* since false positives are envisioned to be less critical than false negatives for future applications to imagined speech.

A trial was considered a *Partial Success* if it captured the majority of the word but clipped either the beginning or the end. Subplots (b), (e), and (h) in Figure 5.3 are examples of *Partial Success* trials. A trial was considered a *Failure* if the word was missed entirely, if the model prediction was erratic or inconsistent, or if a portion of the word was missed from an otherwise well-placed detection.

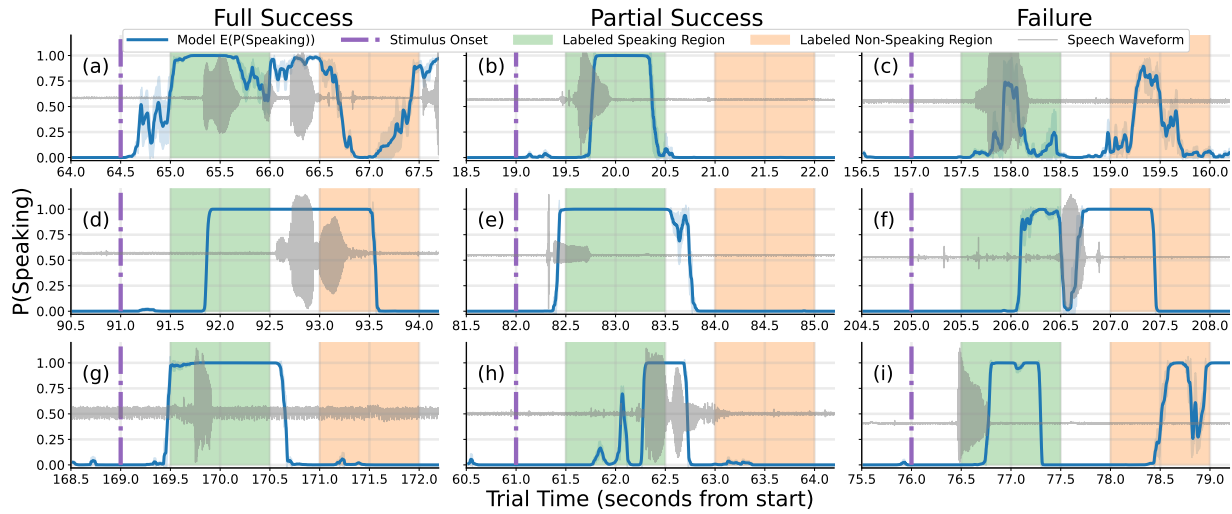


FIGURE 5.3: SincIEEG model predictions of 9 representative words, grouped into 3 categories detailed in Section V.A. The grey trace is the audio waveform from the microphone and represents the participants’ utterances during the word trial. The blue trace, and associated shading, represent the moving average and standard deviation of the model-derived ‘speaking’ likelihood over the previous 15 samples. The green shaded area represents the region labeled ‘speaking’, and the orange shaded area represents the region labeled ‘not-speaking’. Top row: Participants 5, 4, 5. Middle row: Participants 1, 3, 2. Bottom row: Participants 3, 1, 2

Subplots (c), (f), and (i) in Figure 5.3 are examples of *Failure* trials.

For each participant’s best model configuration, the model with the best cross-validation performance was selected and its test-set predictions were assessed using the criteria described above.

Table 5.1 shows the proportion of words assigned to each category for a 115 word test set for each participant for the respective best model configuration. Participant 1 and 2 models were able to very consistently predict speech before speech onset, suggesting that the model and electrode location combination may capture aspects of speech planning. Participant 3 and 4 models had a majority of partial successes. These trials largely exhibited clipping the beginning portion of words, suggesting that the model may be capturing aspects of speech production rather than speech planning.

TABLE 5.1: Prediction Success Over Trials

Participant	Full Success	Partial Success	Failure
1	93 (81%)	11 (10%)	11 (10%)
2	98 (85%)	10 (9%)	7 (6%)
3	36 (31%)	53 (46%)	26 (23%)
4	43 (37%)	51 (44%)	21 (18%)
5	64 (56%)	37 (32%)	14 (12%)

5.5.2 Spectral Band Convergence

Figure 5.4 shows a representative example of spectral bands converging over training epochs. While there was a significant amount of variability in the plots across participants and configurations, there are several consistent observations. First, there is a distinct and consistent difference in the band evolutions during training when dropout is included in the model. With dropout, bands tended to converge more smoothly, rather than exhibiting large jumps in value as observed without dropout. With shared parameters, zeroing a sensor channel

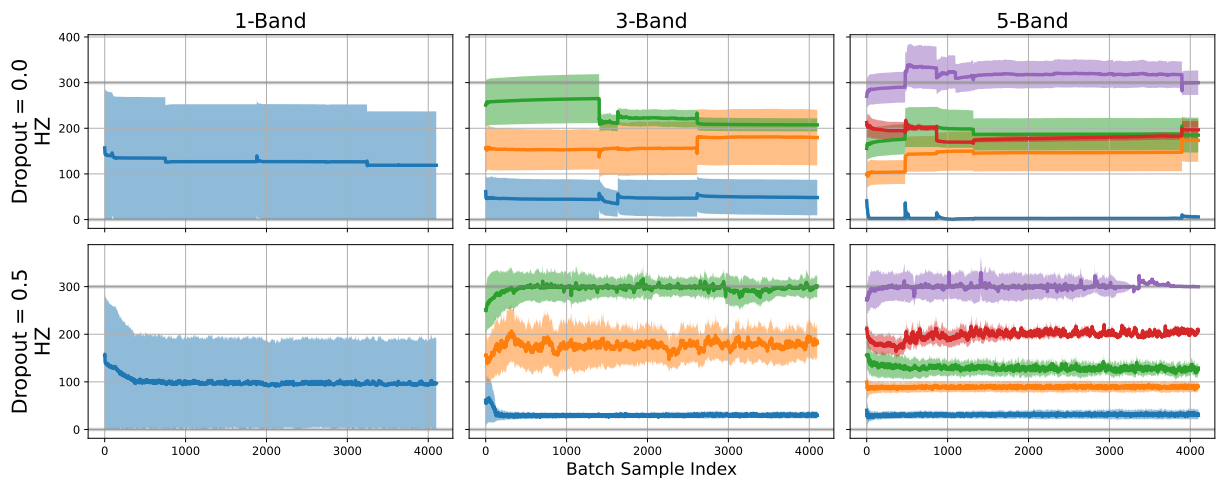


FIGURE 5.4: Spectral band convergence of the 1-, 3-, and 5-band SincIEEG networks for Participant 2. The bold lines are the center of the band, and the shaded regions in the corresponding color are the band bounds. The top row is without dropout, and the bottom row is with dropout.

eliminates its influence and subsequently allows other sensors of varying magnitudes to drive parameter updates. Furthermore, zeroed sensors bias downstream normalization layer statistics towards zero. It is posited that these aspects result in the higher variance stochastic search of frequencies illustrated in Figure 5.4.

The final bands learned for each participant, aggregated across sessions and model configurations, are shown in Figure 5.5, with the bands aggregated across participants shown in Figure 5.6. For better visualization, only SincIEEG models with performance in the top 50% for each participant are included in the figures. The bands are superimposed on a single frequency spectrum as a density plot at high transparency. Each band is plotted in a different color, with more saturated hues representing frequencies common across more participants and model configurations than less saturated hues. This provides a compact conceptualization of the final converged frequencies across models.

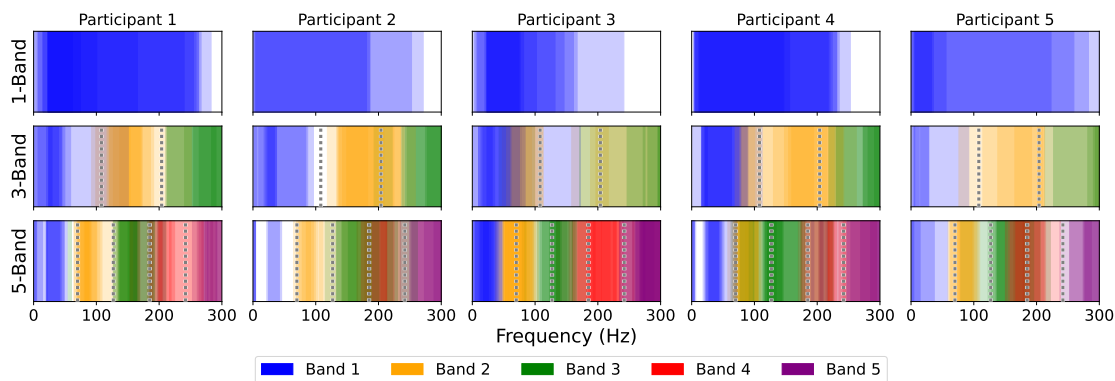


FIGURE 5.5: Learned frequency bands for each participant and 1-, 3-, or 5-band configurations. The selected bands are superimposed on a single frequency spectrum as a density plot at high transparency. Each band is plotted in a different hue: blue, yellow, green, red, and purple. More saturated hues represent frequencies common across a greater number of model configurations than less saturated hues. Vertical dashed lines correspond to the initial cut-off frequencies of adjacent bands prior to convergence. More details on the band initialization procedure can be found in Section 5.3.1.

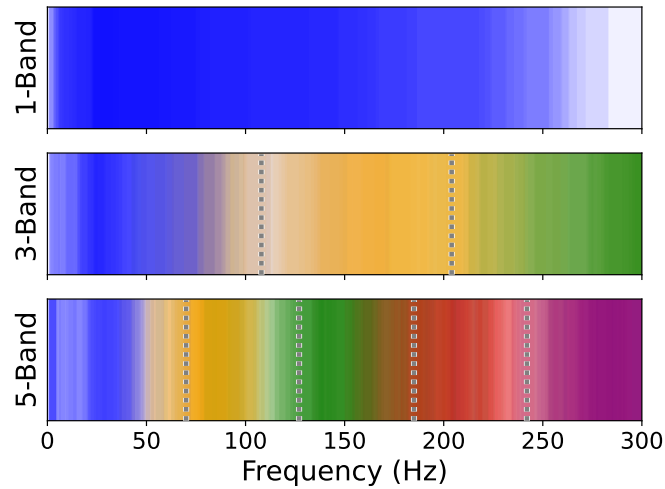


FIGURE 5.6: Learned frequency bands for the top-50% of model configurations across participants for each 1-, 3-, or 5-band configuration, as described in Figure 5.5. For improved visualization, the figure only includes the top-50% of model configurations of each the participants' sessions.

For the 1-band case, the general tendency is for the band to be broad. However, the aggregated data shows that the bands commonly overlapped around 25-75 Hz, implying the lower frequency band may be more predictive than high gamma for the task, as supported by [121].

The 3-band case indicates one lower-frequency band in a narrow range from 20-40 Hz, a broader middle band roughly spanning 120-200 Hz, and a high frequency band converging above 250 Hz. The 5-band case shows similar bands at the low and high ends of the spectrum, with intermediate bands centered at approximately 75 Hz, 150 Hz, and 200 Hz, respectively.

A benefit of the interpretability of learning frequency bands is that the results can be directly compared to known physiologically-relevant bands. Kanas et al. examined 8 Hz wide frequency bands from 0 to 248 Hz, and produced a histogram that ranked bins by contribution to speech detection [77]. It is a multimodal distribution, with two larger peaks, one spanning 0-40 Hz and one 180-200 Hz, with two smaller, broader peaks in the intermediate frequencies.

The 3- and 5-band plots mirror this trend. In the 3-band version, the lower frequency band at 40 Hz and the middle band covering the 150-200 Hz range coincide quite closely with the peaks in the Kanas et. al. histogram. The 5-band version is even more compelling, with the first band again centering on 40 Hz, the two middle bands covering areas around 100 Hz and in the middle hundreds, and the fourth band centering directly at 200 Hz.

5.5.3 Comparison and Benchmarks

Table 5.2 shows the performance of all validation measures in comparison to SincIEEG. The SincIEEG and SincIEEG Non-Auditory results are the mean test fold accuracy for each participants’ best-performing model configuration, effectively the highest bar for each participant in Figure 5.2. Excluding the auditory cortex electrodes did not significantly impact model performance. The causal formulation of the model, and accurate capture of speech onset within the predicted speech window, provides a strong indication that perception of speech was not a driver of the model classification accuracy. The CNN architecture performance is overall on par with SincIEEG. This shows that the interpretable and parsimonious architecture of the SincNet does not compromise model performance.

TABLE 5.2: Model Accuracy Comparison

Participant	SincIEEG	SincIEEG Non-Auditory	CNN	SincIEEG 3-Band LDA	SincIEEG 3-Band SVM	Broadband LDA	Broadband SVM
1	0.939	0.930	0.941	0.748	0.807	0.735	0.726
2	0.979	0.977	0.983	0.900	0.888	0.832	0.827
3	0.957	0.862	0.932	0.876	0.849	0.811	0.794
4	0.893	0.827	0.885	0.743	0.773	0.728	0.713
5	0.941	0.883	0.941	0.710	0.714	0.695	0.692
Mean	0.942	0.896	0.936	0.796	0.806	0.760	0.751

The bands identified by the 3-band SincIEEG for each participant were compared to a broadband approach and classified with LDA and SVM. For both classifiers across participants, using learned bands instead of the broadband showed an improvement in classification accuracy. This implies that SincIEEG provides unique and relevant features due to the participant-specific, empirical, and/or parsimonious nature of the learned SincIEEG bands.

It should be noted that, regardless of whether using learned bands or broadband, the LDA and SVM classifiers with the preprocessed ECoG signals did not achieve better results than SincIEEG. Additionally, SincIEEG was able to achieve better results with greater time-domain resolution than the methods using the preprocessed features.

5.6 Discussion

This work introduces SincIEEG, a deep learning model with an interpretable architecture. SincIEEG is capable of detecting overt speech using unprocessed ECoG recordings based on a diversity of electrode coverage. SincIEEG meets or exceeds the performance of other ECoG speech detectors, with several additional advantages.

In prior work on using ECoG for speech activity detection, Kanas et. al achieved maximum accuracies of 92% [78], and 98.8% with non deep learning classifiers[77]. Other studies used the detection model as part of a larger speech decoding analysis and so did not report specific results on speech detection performance [110, 111]. In comparison to SincIEEG, which uses unprocessed ECoG recordings, these approaches require appreciable signal preprocessing prior to

speech detection. Since the feature extraction is inherent in SincIEEG, any latency introduced via explicit, potentially suboptimal, data-independent preprocessing is mitigated in the processing pipeline - which is critical for real-time implementation.

The architecture of SincIEEG is CNN-based, like that of the foundational work of EEGNet, which showed the viability of CNNs for several tasks using non-invasive EEG signals[89]. The EEGNet architecture was subsequently extended for application in a movement task to intracranial signals, including the addition of a spatial component[119]. This approach is also capable of determining data-driven frequency features, albeit in a manner distinct from SincIEEG. While it is demonstrated that SincIEEG is capable of speech activity detection from ECoG signals, the original implementation was used for acoustic speech detection [128], and it has also been applied to EMG signals [147]. Using a related approach for seizure detection using non-invasive EEG, Fukumori et al. showed that a data-driven approach was superior to static filter banks [44]. Such models that learn the task-relevant spectral bands can be applied to other domains where frequency analysis is central. This is mainly due to the utility of learning bandpass filters, and the flexibility of the scope on which different filters can be learned.

In terms of interpretability, visualization of the learned bands provides a unique modality for studying the relevant spectral features. One consistent observation is that, across all 1-, 3-, or 5-band models and all participants, a low frequency component was always included. This supports prior work that suggests lower frequency features can play a key role in speech detection in addition to broadband gamma [77, 112]. While the present analysis did not attempt to specifically identify the subset of electrodes related to speech production processes, due to the consistent performance results regardless of the hemisphere

of the implant, it is expected that the contributions are largely from the ventral primary motor cortex as shown in prior work [19, 23, 27, 56].

Beyond interpretability, the flexibility of the SincNet architecture's ability to learn different combinations of relevant frequency bands make it promising for implementing transfer learning to leverage existing data for the development and training of generalizable models. Gathering sufficient data and learning robust models for new participants is challenging, particularly for intracranial recordings where available data is limited and the electrode locations are generally sparse and not consistent across participants. In this context, transfer learning can be used to refine the model on a new participant's data after having learned its initial parameters from other participants' data - which can significantly reduce training time and improve model robustness and performance.

Because SincIEEG is capable of learning task-relevant spectral bands across multiple participants independent of precise electrode locations, it has the potential to learn generalized bands for brain regions sampled by the population of electrodes across participants. Furthermore, specific bands can be learned for channel context labels, such as in which brain region an electrode resides. This allows for encoding a spatial component to the transfer learning, initializing different bands dependent on electrode location.

Ultimately, toward the development of a practical speech neuroprosthetic, future work must examine the efficacy of SincIEEG on transfer learning and, moreover, on imagined speech and integration with the subsequent speech decoding pipeline.

"All models are wrong, but some are useful."

George Box

6

Quantifying the Effects of Transfer Learning on Speech BCI Model Performance

6.1 Introduction

Research for the development of speech neuroprostheses using invasive BCI is commonly conducted in a clinical setting. Participants are volunteers and arise solely from clinical necessity, and time with participants is sparse. Transfer

learning is a method commonly used in deep learning which increases training efficiency by leveraging already trained models to help train new models. With sparse data for invasive BCI, there is a need to make model training as efficient as possible.

While some studies have included the technique in portions of analyses, transfer learning remains understudied for invasive BCI. Here, the impact of factors effecting transfer learning on SincIEEG, the architecture described in Chapter 5, is systematically quantified. Descriptive results of transfer learning, both positive and negative, are useful. In order to be successful, transfer learning requires a degree of commonality in the underlying phenomenon, brain dynamics, which may or may not exist. More information about the manner and conditions in which features can be transferred would aid in the creation of larger pretrained BCI deep-learning models.

Transfer learning is examined using two intracranial EEG datasets. One dataset is used to quantify between-participant but within-task effects, where features are transferred from one set of participants to an evaluation participant on the task of overt speech activity detection. The other dataset is used to quantify within-participant but between-task effects, where only one participant is considered, and features are transferred between the tasks of overt speech and imagined speech. The factors other than participant and task examined for their effects on transfer learning are the number of features transferred between models, the source model of those features, and whether or not the transferred features are allowed to update during training. Configurations are tested with a common transferred learning protocol to be more directly comparable, and factors combined in a partial factorial design of experiments. The results are evaluated qualitatively, and quantitatively with statistical tests of significance and regression analysis.

6.2 Background

6.2.1 Transfer Learning

Broadly defined, Transfer learning is a technique in which model parameters from a previously trained machine learning model are transferred to a new model in its initialization. The goal is to improve the training efficiency and performance of the new model, particularly when available data is limited, using knowledge transferred from an adjacent task. For neural networks, transfer learning entails transferring the weights of network layers.

The utility of transfer learning for deep learning was first shown on a large scale in the computer vision domain, and can serve as a motivating example of the concept. With increased computing capabilities and larger datasets, more complex models can be trained. However, for niche computer vision tasks such as ones intended for medical application, data is comparatively sparse. In order to leverage the significant compute resources already spent training large models on more general image data, weights from the pretrained models are transferred to the new model, with a new fully connected final layer that corresponds to the classes of the niche vision task. The model is then initialized and trained with the transferred weights rather than randomly initialized weights. This process is called fine-tuning. Using this method large models trained on ImageNet data [84, 136] have transferred and applied to the tasks of classifying histology [141], brain tumor segmentation [157], 3D medical image generation [30], and classifying radiology scans [39].

Conceptually, the early layers of a CNN are thought to learn general features corresponding to image primitives like edges and shapes, with later layers learning increasingly more abstract feature concepts. Thus, the idea is that treating

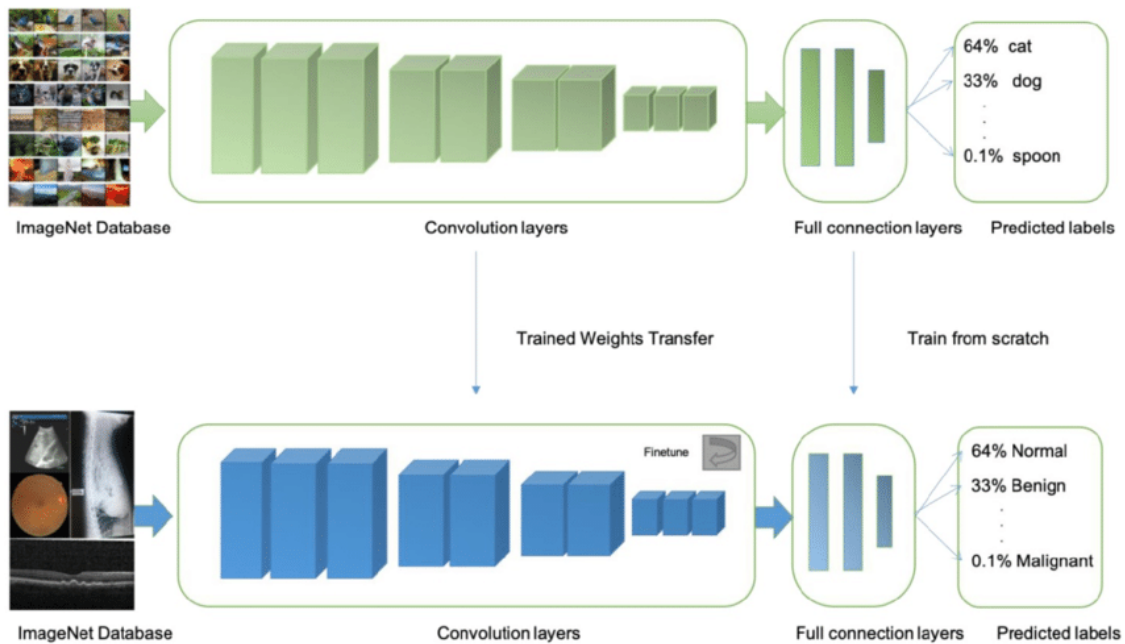


FIGURE 6.1: Transfer learning diagram. Weights from one pre-trained model are transferred. Early CNN layers learn image features, which can be transferred and used to classify images never included in the initial model, with significantly less training time [125]

the early layers of the network as a low-level feature extractor that can then be fine-tuned on many different image classes. Such a design uses comparatively little data because the network does not have to learn the image primitives from scratch - it only needs to label and associate already learned primitives. This concept and the general transfer learning paradigm are illustrated in Figure 6.1.

Current state-of-the-art language models have hundreds of billions of parameters, are trained on data corpora orders of magnitude larger than what is available for other domains, and require considerable financial and computational resources to train [21, 37, 124, 145]. This trend in has made transfer learning from large pretrained models a standard in natural language processing, and increasingly in adjacent deep learning domains [126].

Freezing Weights Versus warm-start

In addition to the question of which weights to transfer, another consideration in fine-tuning paradigms is how the transferred weights will behave during training. As discussed in Section 2.3, training a network involves a forward pass, evaluation of the loss function, and then updating of network weights through backpropagation. When weights are frozen, they are used in the forward pass, but excluded from update during backpropagation. Thus they remain unchanged during training. In contrast to this, transferring weights, but leaving them unfrozen, is referred to as warm-starting. The weights are still updated during training. In this context, the transferred weights take the role of providing the network an informative prior from which to begin training, rather than a naive one.

Generally, the advantages of freezing weights is that there are fewer parameters to update, which improves training efficiency [162]. For particularly large models such as the large language models (LLM) mentioned above, the number of weights makes the computational cost of updating weights significant. In addition, data are a training resource and many niche applications or tasks do not have the data corpus necessary to support the training of all weights for large models. When dealing with large pretrained models, these reasons make any approach other than freezing weights prohibitive under most circumstances, .

In contrast, what stands to be gained from warm-starting and training all weights is potentially better model performance. By freezing weights, the model is not optimizing across the entire possible latent space according to the new task-specific data. If the data underlying the origin task and the target task are similar, then this is less of a concern. As divergence between task data increases,

the transferred weights may be less applicable, and thus benefit more from updating during training time. If there are sufficient data and computational resources for parameters to reach convergence, warm-starting can improve model performance over freezing. The cost to performance trade-off is a consideration to be optimized on an application-specific basis.

6.3 Methods

This analysis characterizes the effect of transfer learning on model performance of the CNN-based speech activity detection model, SincIEEG, presented in Chapter 5. Several dimensions of comparison are considered: (1) transferring weights between participants on the same task, (2) transferring weights between related tasks on the same participant, (3) freezing weights versus warm-starting, (4) the effect of the number of layers transferred, and (5) the effect of transferring from models trained on one, or more, participants. The factors considered are studied across two datasets, using a partial factorial design of experiments. The details of the comparison protocol and experiment designs are detailed in Sections [6.3.3](#) & [6.3.4](#).

6.3.1 Datasets

Both data sets introduced in Chapter 3 and referenced below are used to quantify transfer learning utility.

Single Word

The Single Word dataset introduced in Section [3.1](#) serves as the data source for the between-participant portion of the analysis. The data preprocessing, and the

labeling scheme for the speech activity detection, remain unchanged from their implementation presented in Chapters 4 & 5.

In order to enable training models on data from multiple participants, a 48-electrode version of the dataset is created. For each participant, 48 electrodes are chosen from the set of electrodes not located near the auditory cortex. See Section 5.4.3 for details on the reasoning of auditory cortex electrode removal. Whenever possible, the electrodes are chosen as a contiguous grid, with coverage of brain areas commonly associated with speech activity and production [22, 98]. All multi-participant pretrained models use the 48-electrode dataset, while all single participant models use the participants' full set of electrodes. Other than the electrode subset, all else remained unchanged.

Harvard Sentences

Whereas the Single Word experiment gathered only overt speech data, the Harvard Sentences experiment has components of overt speech, mouthed speech, and imagined speech. Thus, the experiment is used to analyze transfer learning from distinct but related tasks, for the same participant. Specifically, overt speech is compared to imagined speech, which represents a current major challenge in the field. The experiment general experiment details are found in Section 3.2.

Labeling of Overt and Imagined Speech. Word start-stop times were previously extracted from the audio signal of the overt speaking task using the WaveSurfer software package [143]. Time samples within word start-stop times are labeled as 'speaking' for the overt speech task. Because no such ground truth exists for imagined speech, the overt speech labels are used as a proxy for imagined speech. The overt speech word start-stop times are referenced to the onset of the overt speech task stimulus cue, which lasts 4 seconds per sentence. The

word start-stop times are then re-referenced to the onset of the imagined speech task stimulus cue, and as with the over speech, time samples inside the word start-stop times are labeled as ‘speaking’. Figure 6.2 shows the audio signal and experimental stimulus cue of four sentences. The blue regions represent word start-stop times identified during the overt speaking task, shifted over to the regions without audio signal.

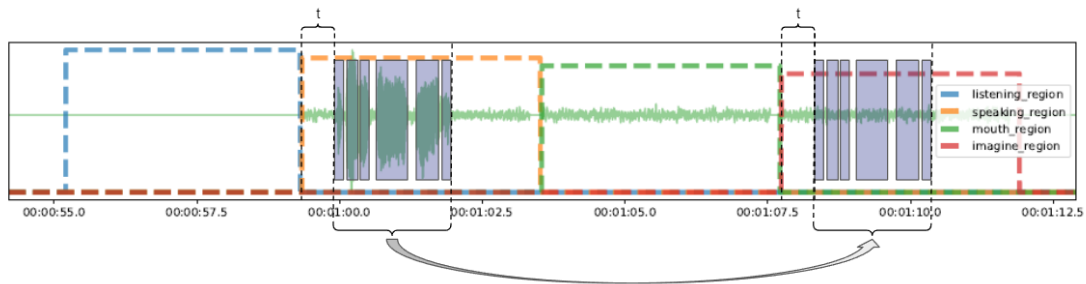


FIGURE 6.2: Labeling scheme for imagined speech task segment of the Harvard Sentences dataset. The transparent rectangles represent word start-stop times previously extracted on the overt speech task. Start-stop times are references to the beginning of the overt speech task segment. Imagined speech is then labeled for the same start-stop times, but in reference to the beginning of the imagined task segment.

6.3.2 Model

The model used for all experiments in the transfer learning protocol is a modified SincIEEG model as presented in Section 5.3 of Chapter 5. In order to restrict the number of factors in model configurations, and for simpler interpretation of the effect of layer transfer on performance, the regularization layers such as dropout and batch normalization are removed from the architecture. Otherwise, the initial sinc layer, and subsequent convolutional layers remain unchanged.

6.3.3 Transfer Learning Protocol

The predominant goal and contribution of this work is the systematic characterization of factors impacting transfer learning of the CNN-based SincIEEG model on the task of speech activity detection. An approach is adopted inspired by a study that quantified transfer learning for CNN image classifiers [167].

All experiments described in Section 6.3.4 follow the same protocol unless specifically stated otherwise. For each participant, a Baseline model is trained to serve as a comparison for the transfer learning experiments. These models are trained for 20 epochs and their best training epoch evaluated on a hold-out test set. This process is repeated 3 times. For each participant, the mean accuracy of the Baseline models represents the benchmark of comparison for the fine-tuning experiments.

Pretrained models are generated to provide weights for transfer. Pretrained models are trained for 10 epochs. For the between-participant transfer analysis using the Single Word dataset, a pretrained model is generated for each combination of held-out participants. That is, for a given participant of the 5 participants in the Single Word experiment, there are 4 one-participant pretrained models, 6 two-participant, 4 three-participant, and 1 four-participant pretrained model, that are trained. Because the Harvard Sentences experiment is not used to explore between-participant transfer, only one pretrained model is generated for each participant.

For fine-tuning models, a model is instantiated for each participant. Then, a transfer learning ‘treatment’ is applied to test a configuration of factors. Factors included the number of layer weights to be transferred, which pretrained model weights are transferred from, and whether weights should be frozen or allowed to update. Once the treatment is applied to the fine-tuning model, it is trained

for 10 epochs, and the corresponding classification accuracy is reported as the primary outcome measure.

In this way, a Baseline model trained for 20 epochs is compared to a fine-tune model trained for 10 epochs with weights transferred from a pretrained model also trained for 10 epochs. The performance comparison is studied between two cases where the total training epochs are the same, in order to characterize the benefit of transferring learning.

6.3.4 Transfer Learning Experiments

The factors explored for their impact on transfer learning performance are summarized in Table 6.1. For both Single Word/Harvard Sentences experiments, the number of model layers transferred and whether transferred weights are frozen or warm-started are analyzed, and the fine-tune model participant.

The quantity of levels for several factors makes a full factorial experimental design prohibitively time-intensive. Thus, a partial factorial design is implemented to fully analyze the factors of greatest impact, and a sub-analysis is performed to characterize factors whose levels are collapsed to reduce total factor configurations. Protocol 1 summarizes the general experiment procedure.

Factor	Factor Levels	
	Single Word	Harvard Sentences
# Layers Transferred	5	5
Freeze/warm-start	2	2
Pretrained Model	15	-
Speaking/Imagining Task	-	2
Participants	5	7
Full Factor Combinations	750	140

TABLE 6.1: Factor Levels and Total Experiment Configurations

Protocol 1: Transfer Learning Experiments

Select: Experiment E_i for $i \in \{hvs, sw\}$, where *hvs* denotes Harvard Sentences, and *sw* denotes Single Word.

if E_{sw} **then**

Select: Participant $p \in \{1, \dots, 5\}$

for $p \in \{1, \dots, 5\}$ **do**

Instantiate fine-tune model $FT(p, PT_p(d_i), l, f)$.

Select: Pretrained model $PT_p(d_i)$, where d_i is the i th model with d donor participants

for $PT_p(d_i)$ **do**

Select: $l \in \{1, \dots, 5\}$, where l is the number of layers to transfer from $PT_p(d_i)$

Transfer l layers from PT to FT

Select: $f \in \{True, False\}$, where if $f = True$ weights will be frozen during training.

if $f = True$ **then**

 Deactivate weights for optimizer

Select: Choose a factor configuration $C = \{f, l, PT_i\}$ where $f \in \{frozen, warm - start\}$ is the training style $l \in \{1, \dots, 5\}$ is the number of layers to transfer PT for $i \in \{1, \dots, 15\}$ is the pretrained model

if E_{sw} **then**

Select: Participant $p \in \{1, \dots, 5\}$

for $p \in \{1, \dots, 5\}$ **do**

Instantiate fine-tune model $FT(p, PT_p(d_i), l, f)$, where $PT_p(d_i)$ is the pretrained model and d_i is the i th model with d donor participants, l is the number of layers to transfer, and if $f = True$ weights will be frozen during training .

Select: pretrained model $PT_p(d_i)$

for $PT_p(d_i)$ **do**

Select: $l \in \{1, \dots, 5\}$

Transfer l layers from PT to FT

Select: $f \in \{True, False\}$

if $f = True$ **then**

 Deactivate transferred weights for optimizer.

Train FT for 10 epochs.

Between Participant - Single Word

For the analysis of transfer learning between participants, an initial sweep of reduced model configurations is trained in order to identify and focus on factors with the largest impact. In order to accomplish this, factors with an ordinal quality are collapsed into a dichotomous factor of their endpoint levels. For example, the *number of layers transferred*, a factor with 5 levels, is only evaluated on the 1- and 5-layer configurations. The *freeze/warm-start* factor, already dichotomous, remains unchanged.

The factor with the largest number of levels is the *number of pretrained participant donors* factor, with 15 possible pretrained models to be used for each participant. Table 6.2 shows an example of the configurations for a participant. To dichotomize this factor, two configurations at each extreme are chosen. Since there is only one 4-donor pretrained model for each participant, this defaulted to the model representing the upper extreme for the number of donor participants. For the lower extreme, a model is chosen at random from the four available 1-donor models in order to reduce sampling bias. This dichotomization results in a 40 model corpus; $5(\text{participants}) \times 2(\text{freeze/warm-start}) \times 2(\text{\# layers transferred}) \times 2(\text{\# pretrained donors})$; for the initial reduced-set factor analysis. Further factor adjudication analyses are performed with either full factor levels, or dichotomized levels as defined here, and are stated in detail in their corresponding results sections.

Pretrained Model Analysis. The pretrained models factor has a number of confounding elements that warrants a more detailed analysis of models their effects on the transfer learning. *Participant* is a known factor of important for model performance. There is merit in teasing apart whether a pretrained model trained on a greater number of participants generalizes and therefore transfers

# Donors	# Configurations	Configurations
1	4	2, 3, 4, 5
2	6	(2,3); (2,4); (2,5); (3,4); (3,5); (4,5)
3	4	(2,3,4); (2,3,5); (2,4,5); (3,4,5)
4	1	(2,3,4,5)

TABLE 6.2: Pretrained model configurations for Participant 1 of the Single Word experiment.

more successfully, or if particular participants' weights transfer more successfully to others.

For this portion of the analysis, a reduced set of the other factors, and participants, is employed in order to reduce computation while maintaining degrees of freedom sufficient to assess effects and interaction effects. The set of participants is reduced to the best (P2), worst (P4), and median(P5) performers for the Baseline models. Because the generalizability of pretrained model weights are of primary interest, the warm-start variation is not considered, and only the 1- and 5-layers of weight transfer are used. This resulted in a 90 model analysis; $15(\text{pretrained model}) \times 3(\text{participant}) \times 2(\text{textit layers transferred})$.

Between Task - Harvard Sentences

Except for the pretrained model factor, the other factors are also studied in the between-task portion of the analysis, serving as additional confirmation of the findings. In addition to *number of layers transferred*, *freeze/warm-start*, and *participant*, the added factor of analysis is *task*. Unlike the Single Word experiment, which only tests overt speech, the Harvard Sentences experiment tests overt and imagined speech. Specifically, sentences are spoken aloud and then the same sentence is imagined with inner speech. Here, the utility of transfer learning within-participant, but between related speech tasks, is assessed.

To accomplish this, two pretrained models are generated for each participant, one using labels corresponding to the imagined speech task, and the other using overt speech labels. The fine-tune models are then trained using the counterpart task of the pretrained model from which weights are transferred. For example, a model that transferred weights from a model pretrained on the imagined speech task would then be fine-tuned on the overt speech task labels. This results in 140 model configurations; 7 (*participant*) \times 2 (*task*) \times 2 (*freeze/warm-start*) \times 5 (textit layers transferred). This represents the full factorial design for the experiment.

6.4 Results

The primary outcome measures analyzed and reported in the results are the model accuracies, and the residuals to the Baseline models. That is, for a fine-tune model, both the absolute accuracy, as well as the deviation of the accuracy with respect to the corresponding Baseline model, are considered. Factor significance is assessed by fitting a multiple linear regression. Categorical variables such as *participant* and training method (*freeze/warm-start*) are codified as dummy variables, comparing categories to a reference category. For these, significance tests the null hypothesis that the treatment category and reference category have the same effect. Trends in data are visualized with swarm and scatter plots.

6.4.1 Between Participant

Table 6.3 shows the grand mean of fine-tune models by participant, as well as the Baseline models for comparison. It serves as reference for latter results and

Participant	Mean Accuracy	Baseline Accuracy
1	81.4%	85.1%
2	90.0%	97.8%
3	89.4%	92.3%
4	67.7%	70.1%
5	75.6%	78.1%

TABLE 6.3: Grand mean of model accuracies across configurations for each participant, compared to the baseline.

a coarse measure of transfer learning utility across all factors.

Multiple Linear Regression shows factor significance

Two initial multiple linear regression models are fit. Both include *participant* as an independent variable, with one using model *accuracy* as the independent variable, and the other instead using *residual accuracy*. Table 6.4 and Table 6.5 report the summaries of the multiple linear regression analysis. Included in the intercept term are *Freeze(False)* and *Participant 1* as default categories, and thus are compared against dummy variables representing the other categories of their factor (e.g. *Participant 2* variable measures the difference between *Participant 1* and the reference category *Participant 1*).

From the comparison of the two tables, it is clear that Participant is a significant factor that impacts fine-tune model performance. That is, when the absolute accuracy is considered, all participant categories are significant. However, when accuracy with respect to Baseline models is considered, all participant categories fail to achieve significance. This implies several relationships between factors. First, the transfer learning procedure does not affect any of the participants disproportionately well or poorly. If this was the case, there would likely be a significant participant category for the residual accuracy regression. Second, when comparing transfer learning effects across participants, residual

	Accuracy			
	Coeff.	S.E.	P> t	Significance
Intercept	1.0061	0.034	0	***
Freeze (True)	-0.1412	0.013	0	***
Participant 2	0.1164	0.018	0	***
Participant 3	0.1022	0.02	0	***
Participant 4	-0.1355	0.02	0	***
Participant 5	-0.0981	0.02	0	***
# Pretrain Donors	-0.0022	0.001	0.109	
# Layers to Transfer	-0.0334	0.004	0	***

TABLE 6.4: Multiple linear regression with absolute accuracy of fine-tune model as the dependent variable.

accuracy must be used because absolute accuracy is substantially affected by the *participant* factor. This is confirmed by the results in Table 6.3, showing a large disparity between average accuracy for Participants 2 and 4.

The *freeze weights* and the *# layers transferred* are significant factors, both with a detrimental effect on accuracy or residual accuracy. *Number of pretrained model donors* follows a different trend. While it does not achieve significance at the 5%, with a p-value of 0.11, the relationship to residual accuracy warrants further review. These factors are investigated in greater detail to further characterize their relationship to transfer learning performance.

	Residual Accuracy			
	Coeff.	S.E.	P> t	Significance
Intercept	0.127	0.034	0	***
Freeze (True)	-0.1412	0.013	0	***
Participant 2	-0.0075	0.006	0.225	
Participant 3	0.0025	0.006	0.688	
Participant 4	0.011	0.006	0.076	*
Participant 5	-0.0054	0.007	0.469	
# Pretrain Donors	-0.0022	0.001	0.109	*
# Layers to Transfer	-0.0334	0.004	0	***

TABLE 6.5: Multiple linear regression with residual accuracy to the Baseline model accuracies reported in Table 6.3 of fine-tune model as the dependent variable.

Number of layers transferred

Figure 6.3 shows the residual accuracies across the numbers of layers transferred from pretrained model, broken out by participant. There is a clear downward trend in accuracy corresponding to increasing the number of layers transferred. However, the variability also increases with the number of transferred layers. Another notable observation is that participants with poorer Baseline performance relative to other participants, such as Participant 4, benefit more from transfer learning. They outperform Baseline models, especially with fewer layers transferred. Conversely, better performing participants, such as Participant 4, have their fine-tune performance negatively impacted relative to Baseline, particularly when more layers are transferred.

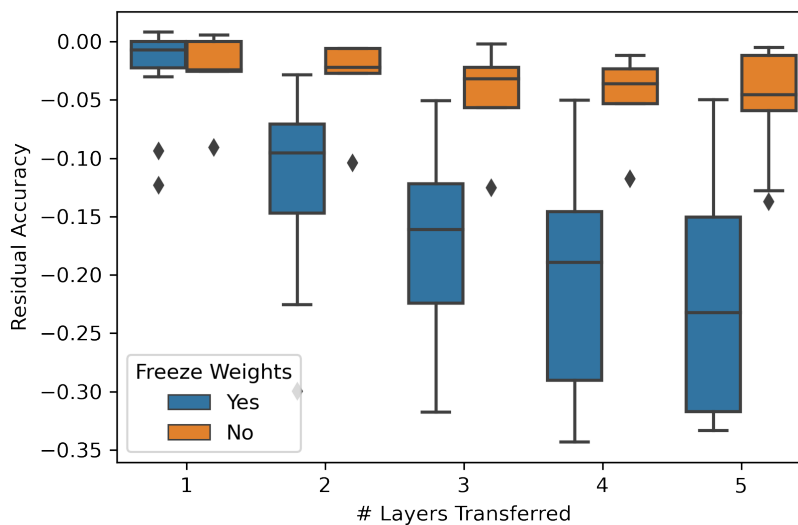


FIGURE 6.3: A box plot of residual accuracy with respect to Baseline models for the Between-Participant experiments, plotted against the number of layers transferred. Boxes are stratified by whether transferred weights are frozen to update, or not, during training.

Pretrained Model Review. The pretrained model sets are the factor with the largest amount of configurations, but do not have a significant impact on

transfer learning performance. Figure 6.4 shows the accuracies of pretrained models for all participants, stratified by the number of participants used in the training of the pretrained model, as shown in Table 6.2.

There is a trend of decreasing performance with an increased number of pre-train donors. The models that include Participant 4 as a donor are highlighted, and show that those models tend to under-perform compared to models that do not have Participant 4 as a donor. This shows that individual participant performance has an impact on the generalizability of models trained with multiple participants.

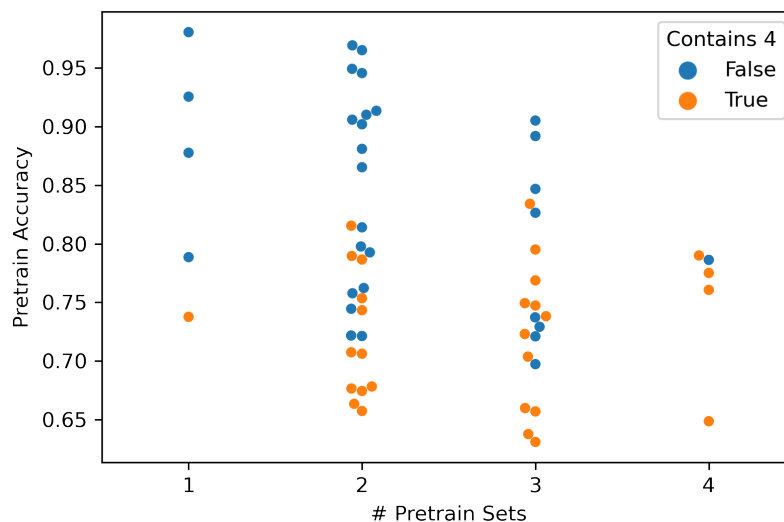


FIGURE 6.4: The number of donor participants plotted against the test accuracy of the pretrained model. Models including Participant 4 are highlighted.

However, when Figure 6.5 is considered, the pretrained model accuracy does not have a significant impact on the final fine-tune model accuracy. The figure shows the fine-tune accuracy for all trials, plotted against the accuracy of the pretrained model from which their weights were transferred. This bolsters the

result of the linear regression, showing that *participant* is the main driver of fine-tune model performance, when considered across all trial configurations.

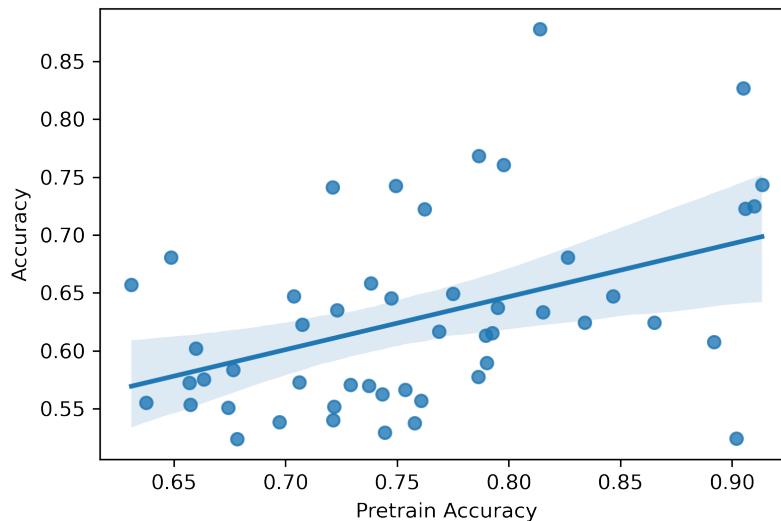


FIGURE 6.5: Pretrained model accuracy plotted against fine-tune accuracy, for all participants.

6.4.2 Between Task

The Between-Task analysis yielded similar trends in results as the Between-Participant analysis. Table 6.6 shows the overall mean accuracy across experiment configuration, in reference to the baseline, for each participant. Each the mean accuracy and the Baseline model accuracies are shown for each speech task; Speaking and Imagining. As a reminder, the Baseline model for each task is 20 training epochs on the same speech task. The transfer learning models transfer between speech tasks, pretraining for 10 epochs, and fine-tuning for another 10 epochs. For example, the mean accuracy for the Imagining fine-tune task is the grand mean of models that have weights transferred from a model pretrained for 10 epochs on Speaking task labels, and then are subsequently fine-tuned on Imagining labels for another 10 epochs.

Participant	Fine-tune Task			
	Imagining		Speaking	
	Mean Accuracy	Baseline Accuracy	Mean Accuracy	Baseline Accuracy
1	68.6%	73.4%	92.3%	89.7%
2	74.9%	77.3%	91.7%	96.9%
3	78.1%	77.5%	87.4%	93.4%
4	70.1%	79.8%	83.1%	89.8%
5	68.0%	67.6%	85.6%	93.2%
6	72.6%	74.8%	88.1%	92.1%
7	72.2%	75.4%	85.7%	95.8%

TABLE 6.6: Grand mean of model accuracies across configurations for each participant for the Harvard Sentences experiment, compared to the baseline. The Imagining Fine-tune Task used weights transferred from the Speaking pretrained model, and the Speaking Fine-tune Task used weights transferred from the Imagining pretrained model.

Both Table 6.6 and Figure 6.6 show the difference between tasks. In the table, the Baseline accuracy for the Speaking task attains good performance, with the lowest accuracy across participants being 90%. Comparatively the Imagining task Baseline models perform worse across all participants. However, with the exception of Participant 5, all participants attain approximately mid-70% accuracies. This is expected, as imagined speech is a more difficult task to decode.

As with the Between-Participant analysis, a multiple linear regression is performed and the results summarized in Table 6.7. The *Participant* and *Freeze Weights Task* factors are codified as dummy variables, with reference categories being *Participant 1*, *Freeze (True)*, and *Task (Speaking)*, respectively. The intercept, *Task*, *Freeze Weights*, and *Layers to Transfer* are all significant factors, whereas *Participant* is not. The coefficients match what is expected, with *Task (Speaking)* conferring approximately 15% benefit, which is seen in Figure 6.6. Freezing weights and transferring more layers both incur a drop in accuracy. Because *Participant* is not a significant factor for the Harvard Sentence dataset, only the regression

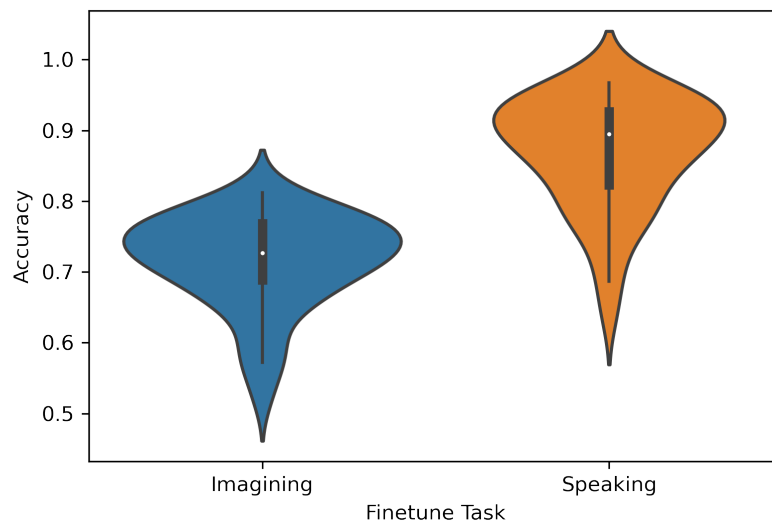


FIGURE 6.6: Model accuracy across participants and configurations, stratified by fine-tune task. Accuracy for the models fine-tuned on the imagining task are 15% lower than models fine-tuned on the speaking task.

with Accuracy as the dependent variable is reported.

Analogous to Figure 6.3, Figure 6.7 shows the residual accuracy for all model configurations for the Between-Task experiment, plotted by the number of layers transferred, stratified by the freeze and warm-start training paradigms. The same trend appears, where the residual accuracy decreases with the number of layers transferred, in the event that weights are frozen. If weights are warm-started and allowed to update, the detrimental effect to accuracy with increased layers transferred is mitigated. Unlike the Between-Participant experiment, which has negligible improvement to Baseline models across all configurations, the Between-Task experiment confers a benefit above baseline in several cases. When the first layer is transferred, a benefit is conferred regardless of whether weights are frozen or not; and for the first three layers transferred, a benefit is conferred if weights are allowed to update.

In Figure 6.8, the data in Figure 6.7 are further stratified not only by *Freeze*

	Accuracy			
	Coeff.	S.E.	P> t	Significance
Intercept	0.8176	0.023	0	***
Task (Speaking)	0.1538	0.012	0	***
Freeze (True)	-0.0862	0.013	0	***
Participant 2	-0.0239	0.026	0.354	
Participant 3	-0.0403	0.024	0.156	
Participant 4	0.0135	0.026	0.599	
Participant 5	-0.0186	0.024	0.435	
Participant 6	0.0252	0.029	0.388	
Participant 7	0.06	0.03	0.093	
# Layers to Transfer	-0.0143	0.004	0.001	***

TABLE 6.7: Multiple linear regression with absolute accuracy of fine-tune model as the dependent variable for the Harvard Sentences task.

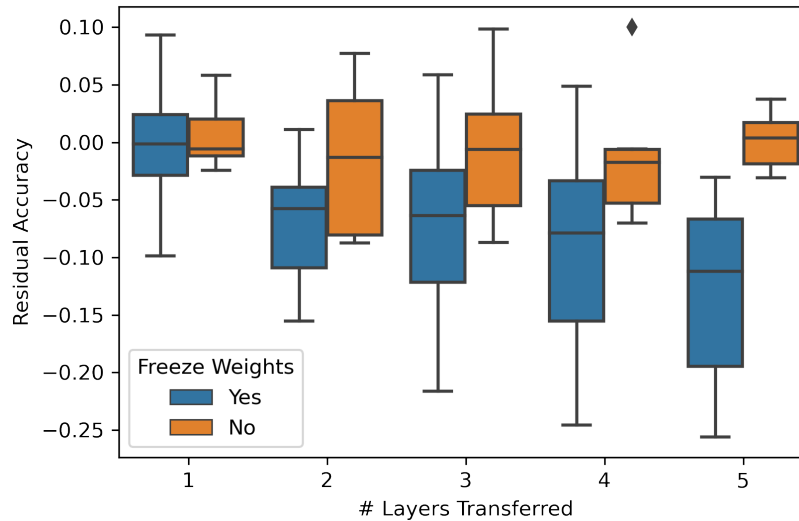


FIGURE 6.7: A box plot of residual accuracy with respect to Baseline models for the Between-Task experiment, plotted against the number of layers transferred. Boxes are stratified by whether transferred weights are frozen to update, or not, during training.

Weights, but also by *Finetune Task*. This shows that while the trend established in Figure 6.7 still holds, the fine-tune task is an important factor. When weights are not frozen (*Freeze Weights: No*), the residual accuracy for both tasks does not decrease as significantly with the number of layers transferred. However, it is

clear that the imagined speech task is the driver of observations where transfer learning confers a benefit above baseline. Similarly, when freezing weights, the imagining speech task decreases less than the overt speech task. It is posited that this is a combined effect; the detrimental effect of more layers transferred is being offset by the benefit that the imagined task receives from transferred layers.

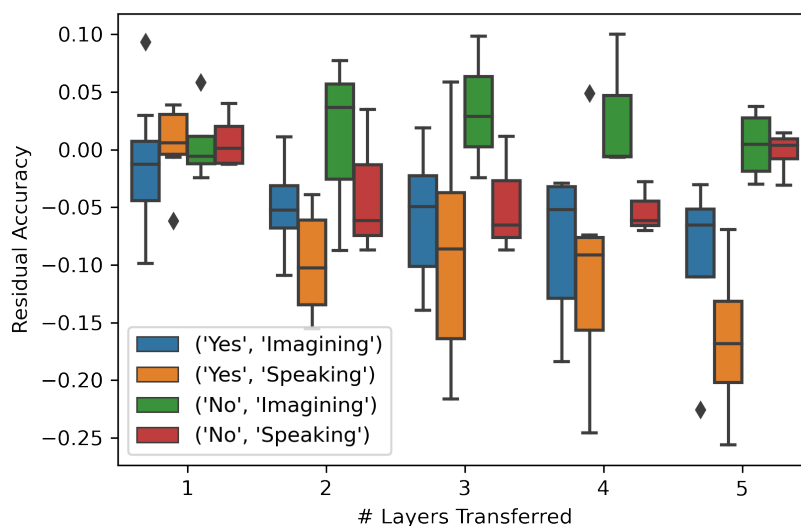


FIGURE 6.8: A box plot of residual accuracy with respect to Baseline models for the Between-Task experiment, plotted against the number of layers transferred. Boxes are stratified by dual criteria; Freeze Weights:{Yes, No}, and Finetune Task:{Speaking, Imagining}.

6.5 Discussion

The effects of transfer learning on speech activity detection was systematically analyzed across a number of factors; split into a Between-Participant experiment using the ECOG Single Word dataset, and a Between-Task experiment using the

sEEG Harvard Sentences dataset. Both experiments and neural signal sensing modalities yield similar results, with a few dataset-specific observations.

In general, for both Between-Participant and Between-Task experiments, freezing weights with few transferred layers is nearly equivalent to warm-starting with any number of transferred layers. The fine-tune training is sufficient to re-converge weights regardless of their initialization (blank or transferred warm-started). Only when many layers (3 or more) are transferred and frozen, is the detrimental effect on performance evident. Figures 6.3, 6.7, and 6.8 show this relationship. This implies to an extent, a lack of commonality in underlying brain dynamics relevant to speech, and that, when weights are allowed to update, these person-specific dynamics drive the convergence of parameters.

However, the computation time is the greatest with all layers transferred and frozen, showing a clear trade-off. The more layers are frozen and transferred, the comparatively larger the detriment to performance. The more weights are allowed to update as with regular training, the less impact is seen on performance (sometimes a mild improvement) but at the cost of compute time. As shown in Figure 6.5, if weights from a well-suited pretrained model are transferred, the drop in overall performance may be worth the saved computation time. In the scenario where all layers are transferred and weights frozen, the pretrained model accuracy correlates to the fine-tune accuracy, though not strongly. This implies that when weights are transferred and frozen, if the pretrained model generalizes better, it provides greater utility to the fine-tune model.

For the Between-Task experiment, there is a disparity in the fine-tune task. The Baseline models for the Speaking task perform better averaged across participants than for the Single Word experiment participants. A possible explanation for this is the more precise labeling scheme utilized for the Harvard Sentences experiment over the Single Word. However, the Imagining task yields

comparatively poor results, but also benefits the most from transfer learning. Intuitively, this makes sense as detecting imagined speech is a harder task than detecting overt speech. The trend indicates that when transferring weights from a difficult task to an easier one (e.g. transferring from Imagined to Speaking) the transfer does not confer a significant benefit. However, if weights are transferred from an easy task to a more difficult, but related, task, the transfer does confer some benefit. This could be due to parameter convergence being bootstrapped towards a 'good' solution that parameters are unable to achieve training on the difficult task alone. This technique has been proposed in other studies [52]. In addition, recent work has linked imagined speech processes to overt speech processes, suggesting they may share features, bolstering the plausibility of performance improvement from transferred weights [121].

These findings corroborate trends and results from prior work in other deep learning domains [167]. Transferring weights can significantly reduce computation time, with an increasing cost to performance. For speech activity detection, when weights are allowed to update during training, quickly performance becomes data-driven and user-specific regardless of the number of weights transferred or their source. Transferring weights between tasks has the potential to be beneficial, but more consideration of task specifics is necessary. At this model complexity, for between-participant transfer, the reduction to computation time is likely not worth the cost to performance. However, transfer learning is likely to show more utility with larger, more complex models that require significantly more computation time to train.

"My drawing was not a picture of a hat. It was a picture of a boa constrictor digesting an elephant."

Antoine de Saint-Exupéry, *The Little Prince*

7

HUBRIS: A Self-Supervised Pretraining Approach for Classifying Disparate Speech Representations from Intracranial Signals

7.1 Introduction

Speech neuroprostheses are designed to decode and synthesize speech directly from the electrical potentials of the brain. However, due to the nature and limitations of the clinical procedures commonly used to obtain research data, existing methods for neural speech decoding generally rely on participant-specific models, trained on labeled experiment tasks. Supervised approaches such as these are naturally restrictive, supporting only one particular participant's sensor configuration and task-related behavior. Instead, self-supervised methods with unlabeled data and explicit handling of sensor configuration may allow for much more flexible paradigms in which multiple participants' data can be pooled for learning general purpose features. Furthermore, methods that learn without labels have broader potential applications, including use in closed-loop online systems in which labels are unreliable or non-existent.

The recent introduction of the transformer architecture ushered in a new era for the deep learning field, showing the attention mechanism to be a simple yet

powerful tool for natural language processing (NLP) and sequence-to-sequence models [153]. The self-attention transformer block served as the foundation for BERT [37] and the GPT series [21], which solidified a trend of self-supervised learning (SSL) where models are pretrained on a large, neutral, data corpus before being fine-tuned on a specific task of narrower scope. More recent vision transformers effectively demonstrate that most data can be treated as a sequence, that self-attention performs as well or better than convolutional neural networks, and that computer vision models can benefit from self-supervised pretraining like their NLP counterparts [40]. Transformers have since been shown to be a viable or superior method for object detection, video action recognition, point cloud shape classification, and multi-modal models [4, 9, 17, 24, 42].

Recently, several studies have explored training language models directly from audio signals rather than text [14, 32, 68]. The key insight of these methods is that, rather than learning a representation in a latent space with continuous targets, they learn from a discretized set of ‘pseudo-speech’ units. Thus, these methods essentially use clustering to learn a self-defined lexicon rather than being constrained to map to an externally defined set such as words, phonemes, or characters. This approach is particularly appealing to speech neuroprosthetic development because it is analogous to the way speech is processed by humans, assigning discrete conceptual meaning to physiological inputs from a persisting audio source, which are also concepts underlying speech production.

In this work, HUBRIS is presented, a sensor-level feature learning methodology that builds on recent progress by utilizing self-supervised pretraining, vector quantization, and spatio-temporal positional encoding for use in speech neuroprosthetics. Semi-supervised NLP techniques are adapted to allow for the pooling of data across participants by re-referencing electrode locations of different participants to a common brain atlas before training. The proposed

framework is used to pretrain a sensor-level feature extraction model on unlabeled data from multiple participants. For evaluation, the pretrained model is used to extract features for an unseen participant's speech related classification tasks. Importantly, the pretrained model's parameters are not updated to accommodate the new participant's data or sensor configuration, forcing the fine-tuning classifier to rely only on the features learned from pooled participant data. Exploratory dimensionality reduction and visualization of the learned features to illustrate class separation for the downstream classification tasks is also performed.

Our results demonstrate that HUBRIS is capable of encoding rich speech representations which can be used for classifying an array of disparate speech-related downstream tasks. These results show promise for a future in which "off-the-shelf" pretrained speech neuroprosthetics models can be used to improve a user's livelihood without the need for extensive data collection and labeling.

7.2 Background

7.2.1 Self-Supervised Learning

Machine learning algorithms can typically be divided into one of three types: supervised learning, unsupervised learning, and reinforcement learning algorithms. Supervised learning, which is the most common form of machine learning, and used in the rest of this work, was introduced in Section [2.3.1](#). Unsupervised learning differs from supervised learning in that there are no labels. Thus, their problem is not framed in the form of finding a mapping from inputs to outputs, because there is 'ground truth' to model predictions to. In this way, unsupervised learning is the identification and analysis of input clusters, anomaly

detection, or principal component analysis [61].

Self-supervised learning is a sub-set of unsupervised learning, which also does not rely on ground truth labels, yet still attempts to solve supervised learning type problems such as classification or regression. Commonly this is accomplished by obscuring a portion of the input data from the model, and, with the entire input serving as ground truth, the model attempts to reconstruct the obscured portions of the signal. This method was used by the BERT and GPT language models [21, 37], by reconstructing masked portions of input sequences. The improvement in performance has made self-supervised learning a standard practice for NLP language modeling [11, 13, 32, 67, 123], and other deep learning sub-domains [4, 31, 130].

The advantage of this method is that training is dependent on input data alone. Without the need for time-intensive labeling, larger corpora of data can be trained on. Data has recently been shown to be even more critical to training successful deep learning models than previously thought [63]. This study on language models showed that a model, Chinchilla, trained on 4 times more data, was able to outperform existing state-of-the-art models on a benchmark task by a significant margin, with only a fraction of the parameters. This indicates that the current data-to-model-complexity balance may need to be re-evaluated in favor of utilizing more data before increasing model size.

7.2.2 Transformers

In Section 2.3 & 4.2.1, feed-forward networks, CNNs, and RNNs were introduced. Each of these architecture types has had a significant impact on the field of deep learning, and represented a significant divergence from existing architecture forms. The Transformer is another such architecture, which was first

presented in 2017 for an NLP application [153]. Since then, it has become the backbone of state-of-the-art models across NLP and other domains, outperforming solely CNN-based architectures on computer vision tasks [9, 40], and RNNs on language modeling [21, 37, 63, 124].

The technique novel to Transformers was self-attention. It is so general an association engine that it allows Transformers to perform well on a broad variety of input data types. Here, an overview of the mechanisms used in the Transformer layer is given. For a more detailed review, refer to the original manuscript [153], as well as The Illustrated Transformer, from which Figure 7.1 was adapted [5].

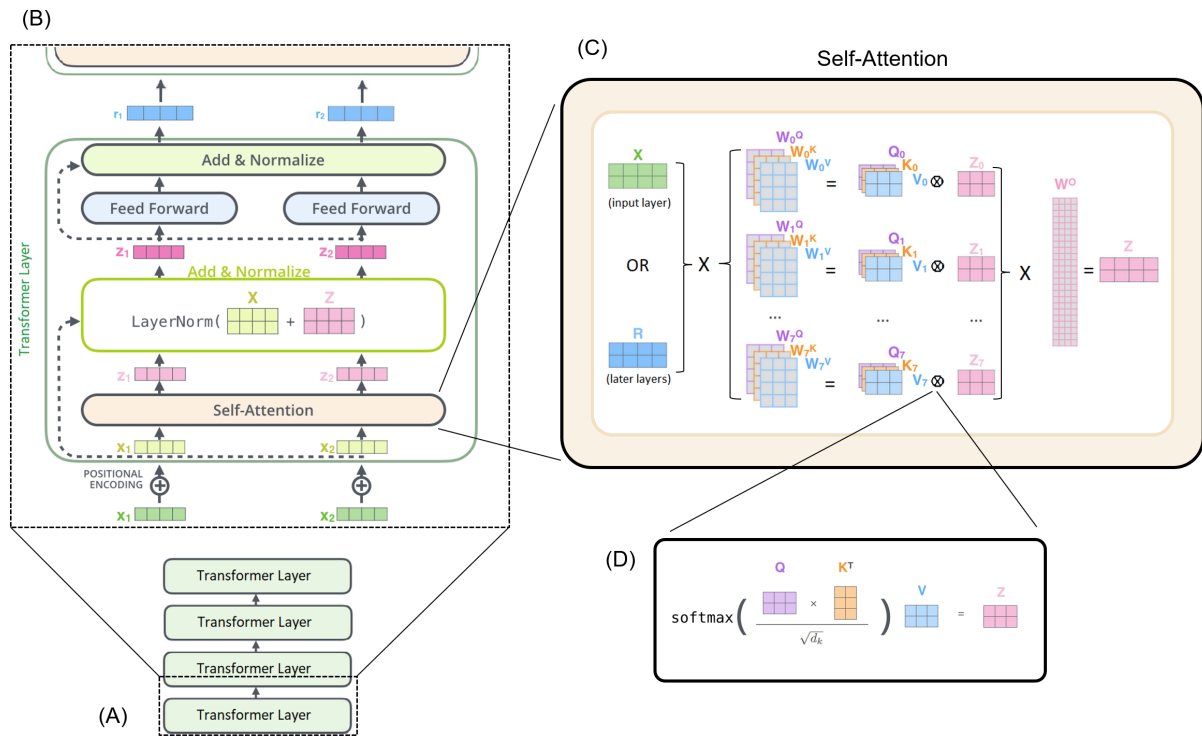


FIGURE 7.1: Diagram of Transformer layer. (A) The backbone of the Transformer encoder is a stack of Transformer layers. (B) Transformer layers are comprised of a self-attention module, and a feed-forward layer, with residual and normalization. (D) The self-attention calculation using the key/query/value arrays. Adapted from [5]

With CNNs, filters of hidden layers deeper in the model encode more abstract features. The same holds for transformer layers. The first layer takes in an input embedding, and outputs context encodings of the same dimension. Later layers take in outputs from the previous layer. Figure 7.1(A) shows transformer layers, (B) is a zoomed in view of the first transformer layer. In the figure, squares represent vectors and matrices. The dimensions are not accurate but are consistent throughout the diagram. Within a transformer layer, inputs are first sent through the self-attention module. Outputs from self-attention and inputs are summed and normalized. Following this, the outputs are sent through a feed-forward layer, and again, residuals are summed and normalized. These are the outputs of the transformer layer, signified in Figure 7.1(B) by the blue vector. The self-attention module applies 3 learnable linear transformations to the inputs, shown in Figure 7.1(C) by W^Q, W^K, W^V . This is repeated n times (8 in the example), in what is called multi-headed attention. This creates n Query/Key/Value matrices. Each Query and Key matrices are multiplied and a softmax applied, and the result multiplied with the Value matrix. The results Z_n are concatenated, and another learnable linear transformation applied, yielding the output of the self-attention module.

7.3 sEEG Dataset - Harvard Sentences

To assess our method, the Harvard Sentences dataset presented in Section 3.2 is utilized. The dataset has tasks that involve listening, overt speech, mouthed speech, and imagined speech, which provides a more challenging dataset for evaluation of the method.

7.3.1 Volumetric Morphing of Electrode Locations to a Common Brain Atlas

Compared to single audio data streams commonly used for NLP and language modeling domains, neural recordings are commonly acquired from tens to hundreds of electrode channels. Additionally, not only is the location of these channels relative to one another important for modeling neural processes, but the absolute channel locations in the brain are also important.

The 3D electrode coordinates reconstructed from CT and MRI imaging data can not be directly compared across participants due to anatomical brain differences. For this reason, each participant's electrode locations were converted from their native brain space coordinates to corresponding locations on the MNI305

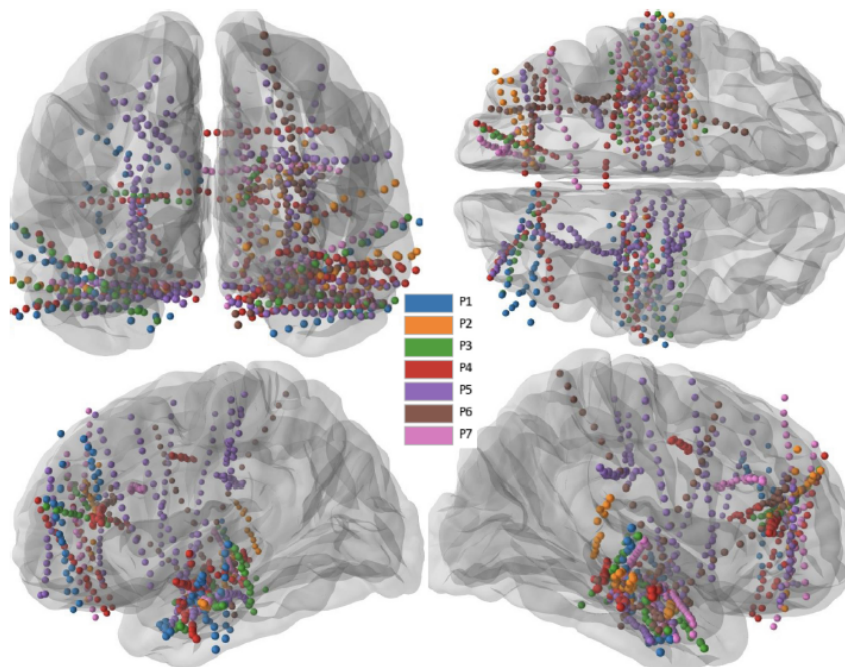


FIGURE 7.2: Common atlas electrode locations for the 7 participants.

common brain atlas [35, 41]. The mapping was done using the Freesurfer software package [43] and MNE-Python python package [49], where further information on the details of the affine transformation procedure can be found [43, 131].

While the MNI brain was selected because it is a widely used common atlas, the critical step is converting the electrodes to a common coordinate space, then any established common atlas can be implemented. This remapping allows sensing locations to be related across participant or even sensor modalities (e.g. ECoG, scalp EEG, etc.), and allows our modeling methodology to leverage the additional spatial information when learning from many participants.

Figure 7.2 shows the locations of all participant electrodes on the common brain atlas. Each electrode is represented using a 3-dimensional vector indicating its location on the common brain atlas. These coordinates are given in the Right-Anterior-Superior (RAS) frame, with positive values in the 3 dimensions referring to right vs. left, anterior vs. posterior, and superior vs inferior, respectively. The coordinate units are in meters, and take on a range of values $[-0.076 \text{ m}, 0.079 \text{ m}]$ across all dimensions. The origin is located at the Anterior Commissure, and the negative y-axis passing through the Posterior Commissure.

7.4 Self-supervised pretraining methodology

Our primary contribution is a model architecture and pretraining methodology for learning generalized feature representations of brain activity, using only unlabeled sensor data pooled from an arbitrary number of participants. This approach is referred to as HUBRIS, and this section describes the underlying model, loss functions, and optimization procedure.

It is shown in Section 7.5 that representations learned by HUBRIS can be used to train classifiers on an array of labeled downstream tasks. Importantly, the HUBRIS pretraining methodology enables fine-tuning on any number of sensors, including new configurations on unseen users.

The model consists of a sensor-level feature encoder, implemented as a convolutional neural network (CNN). The feature encoder's outputs are then passed to a transformer network that learns a latent context vector representation of the input sEEG signal. During the pretraining phase, the model is tasked with reconstructing masked regions of the input signal's latent representations, using self-supervised techniques pioneered by language models [14, 21, 37, 67]. The training is aided by a vector quantization module that discretizes the targets, thus guiding the network to learn hidden units. RAS coordinates are used to learn a spatio-temporal embedding that is added to the input of the context model. The resulting sensor-level model can then be used for feature extraction in a task-specific fine-tuning procedure.

7.4.1 Model Architecture

The HUBRIS architecture is based on the wav2vec2 audio modeling architecture [14], but with significant modifications to support the modality of intracranial sensor data, including changes to the feature encoder CNN, positional embedding paradigm, codebook configuration, and context network size. This section first overviews the input data and the key processing steps across the model's components. Further details on how HUBRIS differs from wav2vec2 are described in each subsection.

HUBRIS's input is an unnormalized 0.5 second segment from a single sEEG channel. The input window is first downsampled to 512 Hz and standardized to

a zero mean and unit variance within the half second window. The segment is

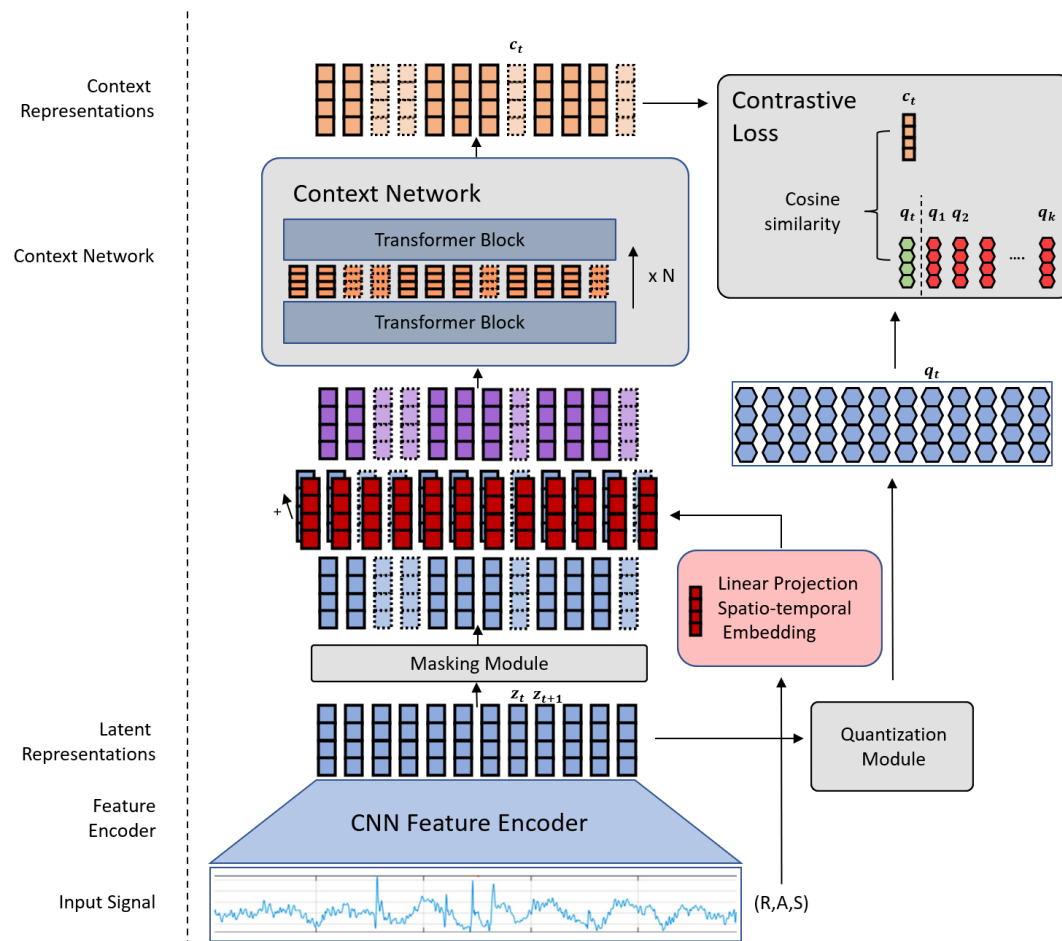


FIGURE 7.3: The HUBRIS pretraining architecture that learns sensor-level representations. A 0.5 s window of normalized sEEG for a single electrode signal is passed to a CNN feature encoder producing latent representations (blue). Spatio-temporal embeddings are created using the 3D RAS coordinates of the electrode (red). The latent representations from the feature encoder are sent to the quantization module. The latent representations are then passed to the masking module, and the positional embedding is added to the masked latent representations (purple). The embedded latent representations are passed to the context network, which is a set of transformer blocks, that finally produce the context representations. The reconstructed context representations corresponding to the masked latent representations are compared to the quantized vectors using cosine similarity in a contrastive loss paradigm.

Further details of each component are in Section 7.4

then passed through a CNN-based feature encoder that generates the latent representations. These latent representations are then passed to both the Quantization Module, where they are discretized into a codebook vector for the objective function, as well as to the context network. The context network is a standard transformer architecture, producing context representations from the codebook distribution. Before entering the context network, regions of context representations across time are masked from the context network by replacing the context representation with a learned mask embedding. Then, spatio-temporal positional information is embedded in the latent representations before being passed to the context model. The masked context representations are learned by having to correctly choose their corresponding quantized latent representation from a set of distractors.

The decision to use a 0.5 s window was driven primarily by prior work, and the intuition that the majority of pertinent information for decoding speech from neural signals will be encapsulated in the neural activity immediately preceding the produced speech. In [82], a speech re-synthesis task was shown to be largely dependent on only 400 ms of neural data centered at the corresponding 400 ms audio signal to be reconstructed, despite the preceding and trailing 400 ms of neural data being included in the predictive model.

Feature Encoder Network

The feature encoder network is used to reduce dimensionality of the input signal before being passed to the Quantization Module and Context Network. The encoder is therefore a 1-D CNN, operating on the fixed length, single-channel, 0.5 s of 512 Hz input sEEG data. The network has 5 convolution layers, each consisting of a 1-D convolution, dropout regularization with probability $p = 0.25$,

layer normalization [10], and a GELU activation function. The first convolutional layer learns 128 filters with width of 7 samples. The next two layers reduce to 64 filters with a smaller 3 sample kernel. The final two layers further reduce dimensionality to 32 filters with a kernel width of 3. All layers use no padding and a stride of 2 to reduce dimensionality. The resulting feature encoding architecture encodes a 0.5 second window of sEEG into 6 sequential steps of 32 channel data (32×6).

Positional Embedding

The original wav2vec2 architecture utilized a grouped convolution relative positional embedding scheme to include temporal position information to the network. Unlike the single-channel audio used in the original design, there is a need to encode the brain signals according to their spatial locations. In order to include not only temporal but also spatial channel information, a positional embedding scheme was implemented that incorporates the electrode RAS coordinates.

The positional embedding used in HUBRIS is produced from a learned transformation of the RAS coordinates described in Section 7.3.1. The first linear layer of the transformation receives the electrode's 3-element RAS coordinates and transforms the input to 32 hidden units. Another 32-unit hidden layer then further transforms the features, before a final output layers produces a 32×6 -dimensional embedding vector. A "Leaky" Rectified Linear Unit (ReLU) with negative slope equal to 0.01 is used as the non-linear transform after each linear layer. A leaky ReLU is used, rather than a standard ReLU, to better handle negative values of the RAS coordinates, while still being computationally simple. The resulting embedding vector is added to the latent representation vector before being passed to the context network.

Quantization Module

The vectors are quantized using a combination of the product quantization [76] and Gumbel Softmax [75] techniques. Product quantization involves creating a set of discrete vectors by defining a number of codebooks G , each with a set of codewords W . Quantization vectors are made by concatenating codewords sampled from each codebook. Thereby a maximum number of quantization vectors is given by W^G . The hyperparameters $G = 2$ and $W = 40$ are assigned, for a maximum possible 1,600 vocabulary size.

Gumbel Softmax enables one-hot encoding of the quantization vectors in a fully differentiable way. A vector of $G * W = 80$ logits are produced for a latent representation vector which after Gumbel Softmax produce one-hot encoding of a word within a group. The quantization vectors are learned via a linear layer, ReLU, and another linear layer which outputs the logits. A diversity loss term, discussed in more detail in the training section, encourages diverse use of the codebook and codewords. This prevents collapse of the codebook, such that it uses only one or few codewords. Details on the exploration of the effect of modulating number of groups and words on a performance of a vector quantized approach are examined in [12].

Masking Procedure

All the latent representations are quantized before the masking step in order to serve as targets for the objective function. The same latent representations from the feature encoder that are passed to the quantization module are also masked before being fed into the context network.

This masking is the basis of the self-supervised learning of the model and is implemented according to [14]. Due to our shorter sequence dimension of

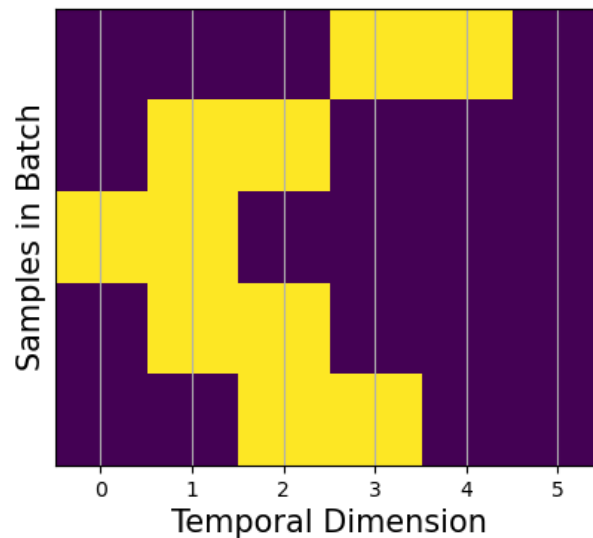


FIGURE 7.4: Illustration of a random mask on hypothetical batch of 5 samples. A model would be required to identify the correct encoding for each of the yellow regions depicted in the figure

only 6 elements, masking is simplified to choosing two consecutive time steps at random.

Each masked latent representation is replaced by the same learnable masking token vector. Overall this results in 1/3 of latent representation vectors masked for the context network. An example of this masking is provided in Figure 7.4.

Context Transformer Network

The context network is a transformer which follows the same architecture as the encoding side [153], also employed by BERT [37], which provides the in-depth details of the Transformer architecture. The proposed context network consists of 6 transformer block layers, each with four attention heads, 2048 feed-forward units, and dropout regularization with $P = 0.25$. The output of each layer is the same dimension as the latent representations fed into the network.

7.4.2 Pretraining

During pretraining, HUBRIS learns speech activity representations from intracranial signals based on an objective function that requires it to correctly identify the true quantized latent representation vector from a set of distractors using the corresponding context representation vector. By using discrete targets rather than continuous vector space targets, the network is influenced towards a parsimonious set of ‘hidden unit’ clusters which represent the underlying speech activity.

Loss Functions

The objective in the pretraining phase is achieved by balancing three loss terms. The first being the contrastive loss function. Given a context representation vector c_t for a masked time step t , the model must choose the correct quantized vector $q_t = QM(z_t)$, which represents the quantization of the latent representation z_t at timestep t , from a set of quantized vectors $q \in Q$ which include itself and K distractors uniformly sampled from other masked timesteps. The loss is calculated by first computing the cosine similarity between context representation vector c_t and quantized vectors Q . The similarity logits are then normalized before taking the negative log of the result for the true vector q_t . All experiments presented in this work use $k = 100$ during pretraining.

$$L_c = -\log \frac{\exp(\text{cosinesim}(c_t, q_t)/\kappa)}{\sum_{q \in Q} \exp(\text{cosinesim}(c_t, q)/\kappa)}$$

This contrastive loss is combined with a diversity loss term. The diversity loss L_d is used to ensure that the use of codewords and codebooks is diverse. The equal use of W codewords from G codebooks is encouraged by maximizing the

entropy of averaged softmax distribution over the codewords for each codebook

\bar{p}_g

$$L_d = \frac{1}{GW} \sum_{g=1}^G \sum_{w=1}^W \bar{p}_{g,w} \log \bar{p}_{g,w}$$

Finally, a feature penalization term L_z is included as the L2-norm of the feature encoder's output. This encourages smaller features and reduces variance.

$$L_z = \sqrt{\sum_{i=1}^{i=N} z_t(i)^2}$$

The final objective function weighs the diversity loss L_d with α , and the L2-norm L_z with λ . Both α and λ can be treated as model hyperparameters during pretraining to help ensure the model converges. All experiments presented in this work use $\alpha = 1$ and $\lambda = 10^{-4}$ during pretraining.

$$L = L_c + \alpha L_d + \lambda L_z$$

Optimization Procedure

Models are pretrained using stochastic gradient descent, with batches of 1,024 sensor windows over 100 epochs. A random 20% of training samples, stratified at the participant-sentence level, are set aside for cross-validation at the end of each epoch during training. The final model is taken from the epoch with the lowest loss L on the cross-validation samples. A learning rate of 0.001 and betas of (0.5, 0.999) were used with the Adam optimizer [80]. The learning rate is reduced by a factor of 0.1 every 10 epochs without improvement on a validation set drawn from the training set.

7.5 Evaluation on Classification Tasks

To assess the viability of HUBRIS, and the generalizability of its learned representations, the features extracted through the feature encoder and context network are applied to three distinct but related downstream classification tasks. These tasks were chosen to be relevant to different aspects of speech decoding; however, they vary in complexity and the components of speech being classified.

For all three classification tasks, 0.5 seconds of sEEG data from all available electrodes is considered, with labels for the half-second window assigned in a task-specific manner. In all cases, classification performance is evaluated using balanced accuracy.

The first classification task is *Speech Activity Detection*. This task is the binary classification of whether a participant is speaking or not-speaking during the half-second window. The second task is *Speech Behavior Recognition*, a multi-class problem of predicting which of 4 speech-related behaviors is being performed: listening, speaking, mouthing, or imagining. The third task is *Word Classification*, where the model must classify which word from a reduced set is being spoken during the window.

7.5.1 Leave-one-participant-out Pretraining

The scarcity of well-labeled intracranial brain data is an important motivation for this work, and with only seven participants, our evaluation must also confront these challenges. A leave-one-participant-out pretraining evaluation method is designed, in which six participants of seven are used for pretraining and a single participant's data is held out for fine-tuning a downstream classifier.

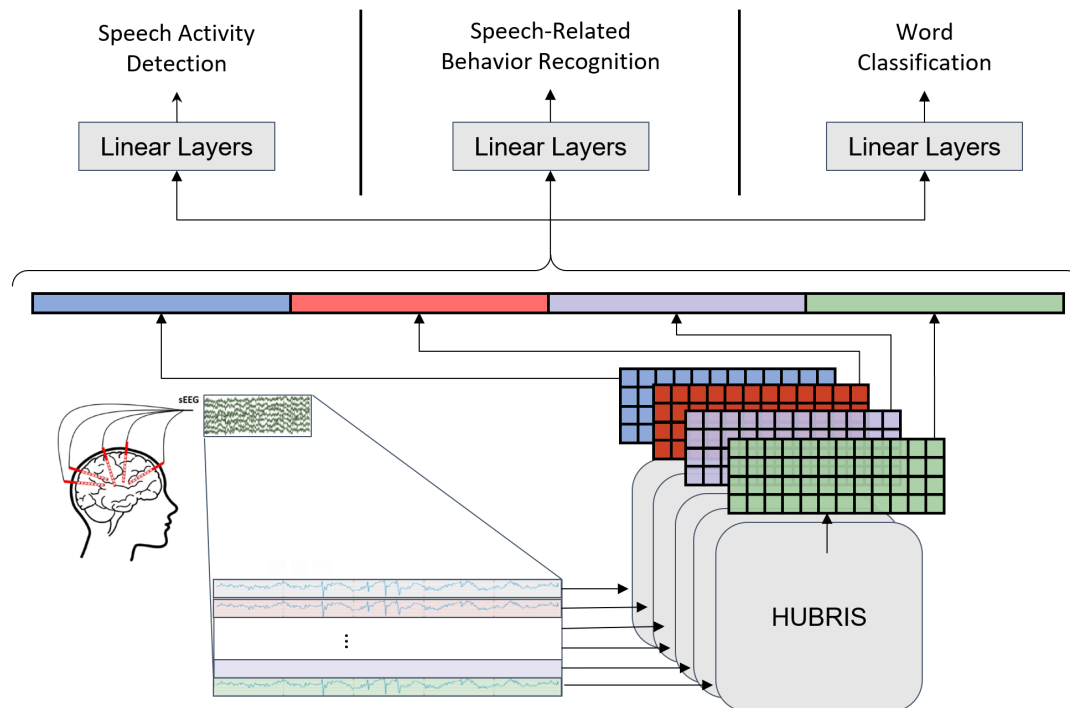


FIGURE 7.5: Diagram of the downstream task training procedure. Given a participant's sEEG signals, a 0.5 s window across all electrodes is processed. The window for each single electrode, and its corresponding RAS coordinates, are passed to a HUBRIS model, producing context representations for each electrode. These representations are flattened, concatenated, then passed through a 16-unit linear layer before finally being passed through the N-class classification output linear layer. The value of N-class is dependent on the task being optimized.

For each participant, that participant's data is excluded and all remaining participants' data is pooled into an unlabeled training dataset. Thus, a unique pretrained model is generated for each participant, one that has never seen a sample from the patient before fine-tuning. This paradigm minimizes data leakage in context feature learning, and ensures the model is not simply memorizing inputs. Additionally, it is intended to simulate the ultimate intended scenario for which a pretrained model based on a larger data corpus is used as the initial model for a new user and subsequently fine tuned. Herein, a pretrained HUBRIS model refers to such a participant-specific, leave-one-out model. All

models employ the same architecture and only differ with respect to the training data.

7.5.2 Downstream Classification

The utility of learned features is assessed by optimizing parsimonious supervised classification models using only the features extracted from HUBRIS. The parameters of the HUBRIS model are frozen, and not updated, to better assess practical applications where new data and available training time are both small. These procedures are referred to interchangeably as fine-tuning or downstream classification.

All three downstream classification tasks follow a similar structure in terms of architecture. Each 0.5 second window of sEEG data is labeled for each of the three tasks, respectively, as described in subsequent sections. To train the downstream tasks, the weights of the entire pretrained model are fixed. For every 0.5 window of labeled sEEG data, every electrode belonging to a participant is passed through the pretrained model in sequence. Every electrode generates the context vector representation of the sEEG input. These representations are flattened and concatenated. This vector, containing the context representations of all electrodes of a participant for a 0.5 window, is then provided to one 16-unit linear layer and a final output linear layer which learns to map to the task-specific classes. The activation function is a leaky ReLU with negative slope of 0.01. Dropout is used with $P = 0.75$ and batch normalization to help regularize the classification optimization.

During fine-tuning, only the additional linear layers and normalization layers are updated. The fine-tuning is performed separately for each participant.

That is, a classifier is trained for each participant on their set of electrodes and corresponding labels.

Speech Activity Detection

For speech activity detection, the audio data is labeled using an energy threshold to generate binary speech/non-speech labels for each segment. Only task segments from the speaking region are processed for speaking labels, but non-speaking labels are taken from any low energy windows in any task region. The sentence narration audio was removed to prevent false-positives in this automatic labeling process. Windows of 0.5 s sEEG data corresponding to overt speech are assigned a *speaking* label. An approximately equivalent quantity of windows with audio below the threshold were assigned a label of *non-speaking*.

Speech-related Behavior Recognition

The behavior recognition task labels each 0.5 s sEEG window according to one of four speech-related behaviors; *listening*, *speaking*, *mouththing*, or *imagining*. The resulting 4-class classification problem challenges the model to disambiguate highly related activities. The experiment protocol codes the regions with associated experimental cues, visualised in Figure 3.3. Labels are assigned to the sEEG data according to these task intervals. Each interval is 4 s in length; however, the initial 0.5 s and the final 1.0 s of the 4-s interval is not labeled to better ensure that the labeled data is representing the speech-related behavior within the interval.

Word Classification

The word classification task requires the fine-tuning model to classify a word from a restricted set. The data collection protocol does not repeat sentences, but

across all sentences there are a set of words that are repeated and are not stop words. Stop words are the most common words such as articles, prepositions, or pronouns, which are commonly excluded when training natural language schemes. Ten such non-stop words are selected arbitrarily for the present analysis.

Forced word alignment was performed on the audio data to identify word start and stop times. These word start-stop times were used to label the corresponding sEEG segments with the associated word.

The training set consists of the sEEG windows corresponding to all 10 selected non-stop words from their first appearance. For the test set, the model is given an sEEG window from 5 of the 10 words, taken from the second appearance of the word. The remaining second appearances of each word are used for cross-validation during training. For example, if the bolded training word was taken from the sentence *The fish turned on the bent **hook***, then the word would be tested on sEEG segments corresponding to the subsequent sentence *He was caught, **hook**, line, and sinker*. In this way, the word classification task is challenged with previously unseen data. The selection of which word's second occurrence is included in the cross-validation versus the test set is randomized for each participant's trial.

7.6 Results

The performance of HUBRIS is evaluated by comparing the balanced accuracy for each of the respective classification tasks. Figure 7.6 and Table ?? show the balanced accuracies of the three tasks for each participant, the overall average accuracy, and the chance accuracy of the classification task. In order to verify

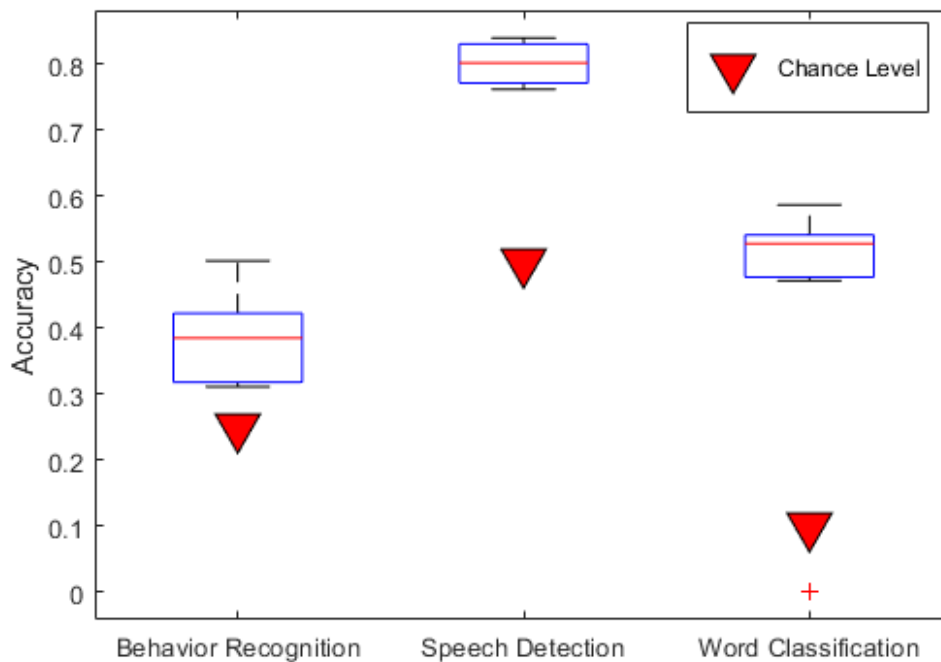


FIGURE 7.6: Box plot of accuracy across participants for the 3 downstream tasks. Red triangles indicate chance-level accuracy for each task.

chance accuracy, the downstream tasks were trained on randomly assigned labels, and these results are included in the table.

Compared to the Speech Activity Detection and the Word Classification task, Speech-Related Behavior Recognition had higher inter-participant variability, and was overall closer to chance accuracy for the task.

The Speech Activity Detection task's average balanced accuracy is 80.2%, and achieves the smallest variance among the tasks. All participants were significantly above chance accuracy of 50%, and the worst performer attained 82.7% accuracy. For comparison, in a recent speech activity detection study using the same Harvard Sentence dataset, logistic regression models as well as CNN models achieved an average accuracy of 82-84%[\[146\]](#). Several other studies using intracranial signals reported results ranging between 80% - 94% accuracy[\[78, 111\]](#). All these studies used fully supervised learning methods.

Participant	Speech-related Behavior Recognition	Speech Activity Detection	Word Classification
1	33.4%	91.1%	-
2	36.2%	95.0%	54.1%
3	44.3%	82.7%	48.3%
4	49.4%	89.3%	40.9%
5	36.1%	88.9%	55.7%
6	46.4%	89.9%	56.0%
7	49.8%	91.7%	62.6%
Average	42.2%	89.8%	52.9%
Random	27.0%	54.8%	12.4%
Chance Acc.	25%	50%	10%

TABLE 7.1: Balanced accuracy of downstream tasks. Participant 1 did not have a complete dataset needed for Word Classification and is therefore omitted.

Word Classification yielded the most promising performance of the three tasks. With only one training example of each word from the repeated word set, average participant accuracy was 52.9% when tested on repeated words. Moreover, the hold-out words were from entirely different sentences with different broader context. As mentioned in Section 3.2, Participant 1 did not complete all 50 sentences during the data collection experiment. They did not have the samples required to be evaluated on the Word Classification task, and thus are excluded from this portion of the evaluation experiments.

A notable observation seen in Figure 7.6 is that, while there were some exceptions, there was a tendency for participants to perform consistently in comparison to other participants across the three tasks. For example, participants 4 and 6 performed in the top half for all tasks, while participant 3 and 7 performed in the bottom half.

Figure 7.7 shows the cross-validation loss during pretraining for all participants. It can be observed that the models converge to generally similar losses,

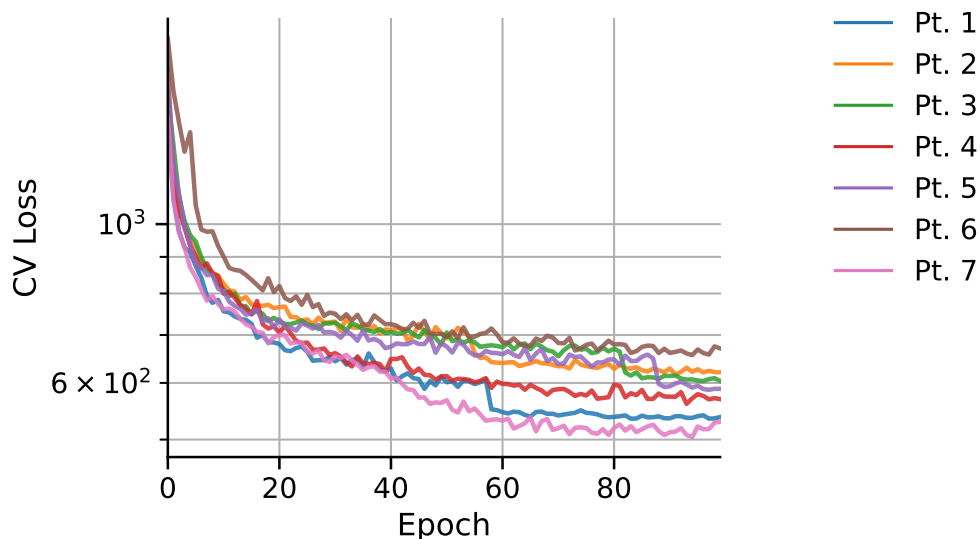


FIGURE 7.7: Cross-validation loss of HUBRIS model over pretraining epochs.

that is, there do not appear to be order-of-magnitude differences. This is expected, as each model shares approximately 6/7 of the electrode data corpus. Nevertheless, it is confirmation that there is some measure of consistency in the convergence process.

The confusion matrices of downstream classification tasks are shown in Figure 7.8. The Behavior Recognition task shows that *imagining* was confused more often with *listening* and *mouthng* than with *speaking*. Further, *speaking* was confused most often with *mouthng*. This observation may indicate a closer mechanistic relationship between imagined speech and listening or mouthng than over speaking [46, 88, 92].

Figures 7.9, 7.10, and 7.11, and respectively show the 3-component t-SNE [99] of the pretrained features for each fine-tuning task. The figures give an indication that the context representations learned by HUBRIS are meaningful to each speech domain task. It is observed that, for each task, there are clear regions of separability for each of the classes. Particularly, word classification in Figure 7.9

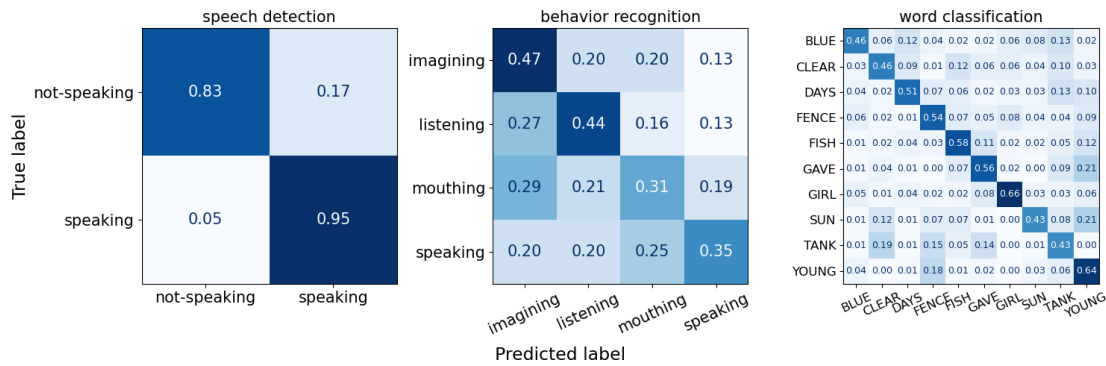


FIGURE 7.8: Confusion matrices of fine-tuning classification tasks across all participant test sets. Each row (true label) is normalized independently, giving the portion predicted class labels across all of the true samples evaluated.

shows distinct differentiations between words. This likely contributes to the impressive performance of the word classification task given comparatively little training data, as the context representations show clear differentiation prior to supervised training.

7.7 Discussion

The performance of HUBRIS on the three disparate downstream tasks showcases the generalizability of the self-supervised features learned by the procedure. While all tasks achieve better than chance accuracy for all participants, in particular, the speech detection task approaches accuracies on par with other supervised learning methods, and the word classification task exhibits promising results using only a small amount of labeled data.

The main objective of this analysis was to develop and establish the efficacy of the pretraining procedure and model, using the performance on downstream tasks as a measure rather than an end goal. The manner in which the model

pretrains inherently makes it difficult to draw conclusions directly from analyzing the context representations, and is further complicated with the addition of the fine-tuning linear layers. Thus, performance on downstream tasks are used to draw indirect evidence of the efficacy of pretrained features. The classification tasks were purposefully selected to cover disparate speech representations that yield a range of classification challenges. Otherwise, the selected classification tasks are somewhat arbitrary with respect to common speech representation available in this particular dataset, and the framework is designed to be agnostic to specific speech representations.

Performance on the Speech-related Behavior Recognition task, while comparatively exhibiting the weakest performance, can also be considered the most

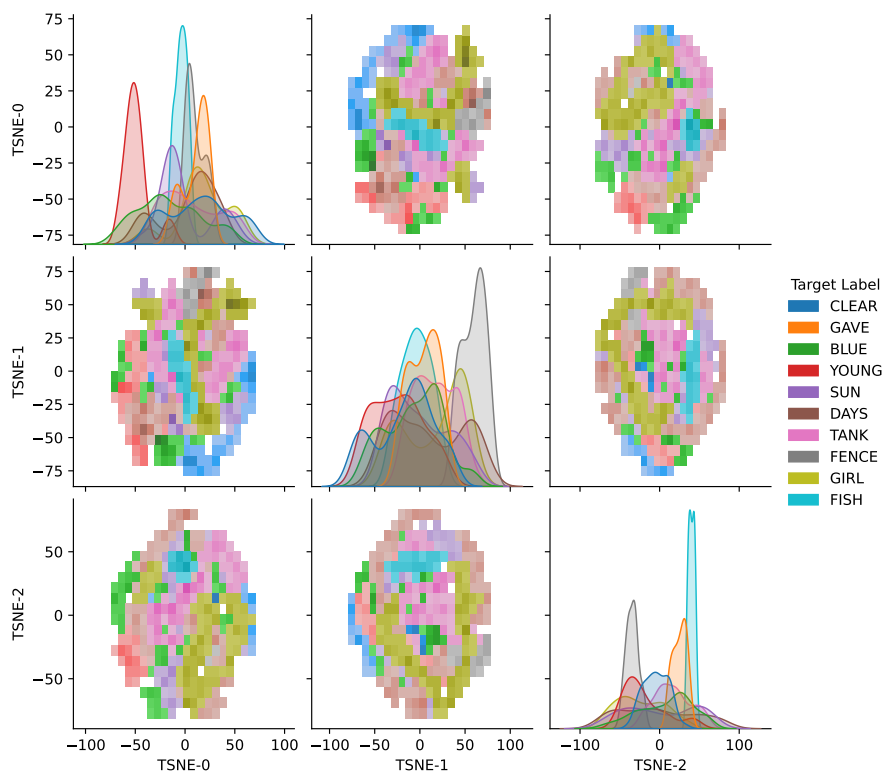


FIGURE 7.9: Visualization of 3 t-SNE components from the pretrained features on an unseen users data (Pt. 7), colored by the **Word Classification** fine-tuning task.

challenging of the three classification tasks. The neural circuits for perceiving speech, and producing overt, mouthed, and imagined speech, are highly intertwined [46, 121, 133]. Nevertheless, it is encouraging that the context representations of the model appear to encode some neural correlates of these behaviors.

The Word Classification task is essentially a few-shot learner, only provided a pair of training examples (i.e. word utterances) of each class before evaluation - one for optimization, and another for validation. In contrast, a study recently showed results ranging from 30-60% on a similar classification task using ECoG signals and a transformer architecture, though in a fully supervised manner[83]. This demonstrates the utility of the self-supervised method: using only unlabeled data, features are learned and guided into hidden, likely sub-word, units. Then, it is posited, comparatively little data is required to map these features to

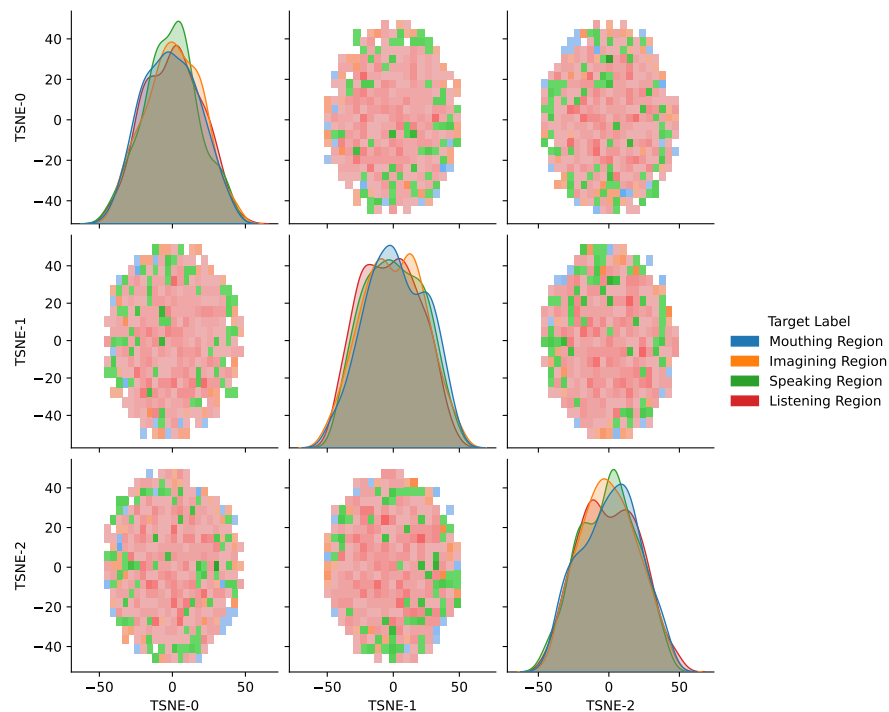


FIGURE 7.10: Visualization of 3 t-SNE components from the pre-trained features on an unseen users data (Pt. 7), colored by the **Behavior Recognition** fine-tuning task.

a word space.

The success of HUBRIS is likely due to several factors. The self-supervised training of latent representations with quantized targets, while keeping the learned context representation as continuous, is a gentle influence to learn not fully-discrete codewords, but instead grouped clusters in the continuous space, known as hidden units. In this way, features are guided towards self-determined clusters, while still allowing the model to fully leverage the rich context of continuous-space features. Because of the self-supervised nature, these clusters are not matched to any linguistic unit, such as words or phonemes, and instead are self-determined by the network. However, because the training data are strictly

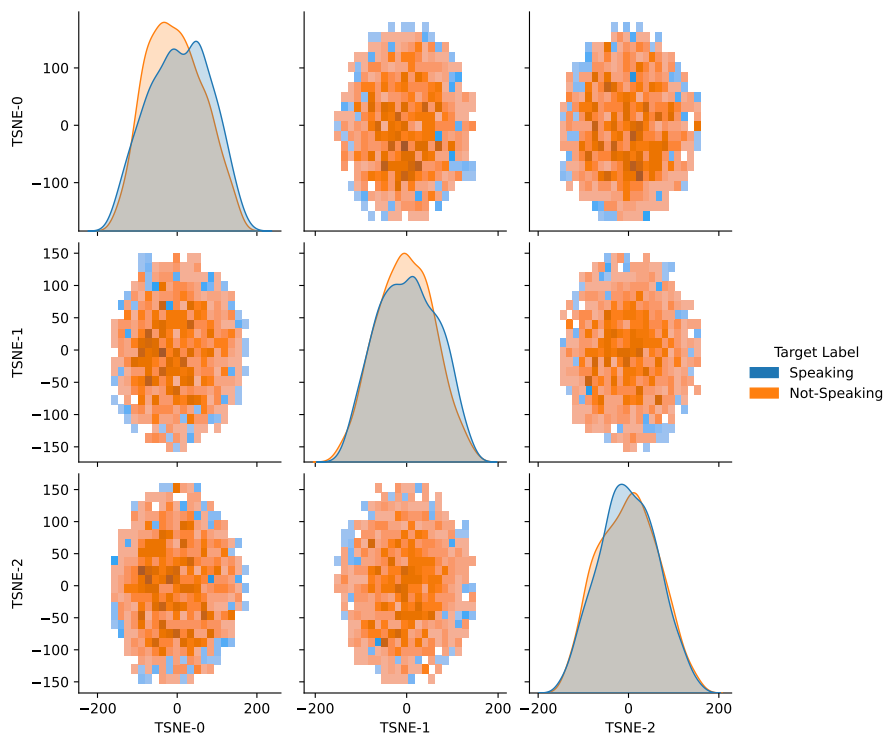


FIGURE 7.11: Visualization of 3 t-SNE components from the pre-trained features on an unseen users data (Pt. 7), colored by the **Speech Detection** fine-tuning task.

from the speech domain, it is likely that the hidden units are converging to neural versions of some, possibly combinations of, linguistic units. This is a potential explanation as to why the Word Classification task was successful using sparse training data.

The projection of RAS electrode coordinates to a common brain atlas allowed for the pooling of data from multiple participants to provide informative absolute brain location data of electrodes to the model. With a sufficient data corpus and electrode coverage, this type of self-supervised model has the potential to train a brain signal regression given neighboring signal data.

During model development, several issues were observed that adversely impacted training success. The objective term weights, α and λ , required exploration with small experiments to find appropriate configurations that avoided codebook collapse - wherein the model used few codewords or the codewords would have little variance overall.

Under some conditions, HUBRIS would fail to converge and maintained at a high CV loss, but this could not be consistently replicated and never occurred with the configuration presented in this work. Large improvements in consistency are found after implementing appropriate weight initialization. Convolution and linear layers were initialized from $\mathcal{N}(0., 0.02)$, BatchNorm parameters from $\mathcal{N}(1., 0.02)$ with a bias of zero, and LayerNorm parameters are initialized with 1.0 and zero bias. This implies a sensitivity to initial conditions and hints at further improvement through more sophisticated initialization schemes and complex learning rate paradigms as explored in other language model methods [12, 32]. This is likely an attribute of the model architecture rather than the particular data.

The number of transformer blocks, and the latent representation vector dimension, and other factors that determined model complexity, impacted performance on downstream tasks. This is likely a balance with the amount of available data. Language models using transformer architectures often have a ‘large’ model variant with 24 transformer blocks [14, 32, 68]; however, these models are typically pretrained using on the order of 60,000 hours of data, whereas the proposed approach was effective using slightly over 1 hour of data for pretraining.

Additional sEEG training data would allow for a deeper model with more transformer blocks, a longer input sequence, or larger embedding dimension, which might in turn provide greater context and learn richer representations of multiple speech and speech related processes. The downstream tasks explored here are constrained by the nature of the speech data available. With enough data, and a sufficient depth of network, it is conceivable for HUBRIS to serve as the backbone of an even more generalized model; one capable of discriminating overt or imagined speech intention, then decoding the speech from the same initial feature set.

As this work is largely an initial proof-of-concept, there are many possibilities to extend and optimize this framework. Here, a linear output layer was implemented for simplicity and comparability; however, more complex decoder paradigms, such as a GPT transformer stack may be better suited to more complex downstream tasks. The recent and growing corpus of publicly available data sets [154] can be leveraged to pool data from participants across experiments, and potentially across sensing paradigms, as long as the dataset includes electrode coordinates for the positional embedding.

This work developed and evaluated HUBRIS, a transformer-based self-supervised model that learns speech-related hidden unit representations from unlabeled sensor-level sEEG data. The outputs of HUBRIS after pretraining are used to

fine-tune a classifier on labeled data from three disparate downstream speech classification tasks. All tasks perform above chance accuracy for all participants, while the speech activity detection and word classification task performance rival competitive supervised learning methods.

'Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.'

Marie Skłodowska-Curie

8

Conclusions and Future Work

In this work, evidence is presented that deep learning is a modeling approach well-suited for the development of adaptable BCI, and that user-centered BCI based on data-derived features is capable of achieving performance that rivals static, preconceived feature extraction methods. The key contributions and impacts are summarized herein.

Chapter 4 focused on the use of unprocessed signals with deep learning architectures, and showed that using researcher-derived features does not yield superior performance on speech activity detection. Chapter 5 further showed

that conventional static feature extraction can neglect task-relevant information. All analyses in this work have used unprocessed signals and were able to achieve results comparable to or exceeding methods that used preconceived feature extraction. This is not surprising, considering that a deep neural network is intended to serve as a universal function approximator. As long as task-relevant information is introduced to the model, it should learn the necessary functional transforms, if they are indeed relevant to predicting the underlying process.

These findings are a compelling contribution to the field with a broad impact. They challenge faulty established conventions concerning feature extraction methods. Adoption of these findings stands to greatly increase the reproducibility of studies, a serious problem in the field.

The results of Chapter 5 showed that data-driven features are person-specific and that, in terms of features, there is less variability intra-person than inter-person. Additionally, results in Chapter 4 similarly showed that generally model performance can be improved by using person-specific features rather than the same preconceived features for all participants. Finally, the transfer learning experiments of Chapter 6 show that transferring weights between participants does not perform better than models trained with participant data. All of these findings corroborate the claim that models using the same static, preconceived features for all people will not perform as well as models that derive data-driven, person-tailored features for each user.

These findings are a novel corpus of results that are an important contribution to the field, as they provide a convincing argument for the use of data-driven modeling as the de-facto best practice in the development of robust speech BCI.

It is known that the location of neural dynamics is critical to speech processes (e.g. activity in Broca's area is not equivalent to activity in the auditory cortex).

Studies often analyze which electrode locations are most strongly implicated in model prediction. In Chapter 7, this location information directly is directly incorporated as model inputs. The addition of electrode locations was an essential part of the methodology that allowed for pooling data across participants.

Between-person transfer learning was shown to be a difficult task in Chapter 6. Yet Chapter 5 and Chapter 6 also show that while features learned from underlying brain dynamics are not entirely disparate between individuals. Transfer learning was then successfully implemented in Chapter 7. Further analyses are needed to fully explore the reason for its success, though anatomical positional embedding and a greater overall model complexity set it apart from the method used in Chapter 6.

The HUBRIS model presented in Chapter 7 also leverages learning from unlabeled data to further increase the corpus of data available for model training. More data is critical for training larger, more sophisticated models such as HUBRIS. The HUBRIS modeling approach uses signal reconstruction techniques to learn a quantized set of vectors, which in turn help learn ‘hidden units’. The units are self-defined clusters pertaining to underlying behaviors, which are, in this case, speech-related. The hidden units are context-rich representations that were successfully used on several downstream speech tasks.

A common theme of this work is to argue that rather than using knowledge to hard-code variables or model parameters, it should be used to gently guide model convergence by exposing the model to relevant information. In this case of modeling speech processes using iEEG signals, this information can be distilled to (1) frequency domain information: the oscillatory properties of the electrode signals summarize the neural dynamics of nearby brain matter; (2) positional information: the anatomical location of where the neural dynamics are occurring; and (3) temporal information: the sequence of neural dynamics over

time.

The HUBRIS model is deliberately designed to combine all three types of information to produce speech representations that generalize well, all while trained on unlabeled and unprocessed data, as well as successfully transferring models between participants.

A modeling paradigm with any one of these advancements would be a significant contribution to the field of speech neuroprosthetics. The HUBRIS implementation combines all three, and represents truly novel work that is well-suited to serve as a basis for the next generation of data-driven speech BCI models.

Future Work

The methods introduced in this work can be extended in several important directions. The ultimate goal of a speech neuroprosthesis is to be able to decode imagined speech in real time. By necessity, such a system capable of either textual decoding or speech synthesis from imagined speech must function in a on-line, closed-loop fashion. While Chapter 6 showed that deep learning models could reliably detect imagined speech, all implementations of this work were offline. A natural extension would be to implement them online in the clinical setting.

In particular, the innovations of HUBRIS in Chapter 7 have the potential to create a paradigm shift in the way that speech neuroprosthetics are trained and developed. With the introduction of self-supervised learning and without the need for labeling, clinical experiments can now leverage passive learning designs. For example, the typical constraints of clinical experiments yield less than an hour of data. In traditional designs, the collected data must be labeled for supervised training for evaluation in the same experimental session. This is generally impractical to achieve due to the aforementioned time constraints. With a

self-supervised model, recording and modeling equipment can be left with the participant to continuously collect and train on data passively, while the participant is engaging in natural speech behaviors outside of the experiment. The model parameters could then be fine-tuned during online experiments. This paradigm would allow for a great deal more data, which would in turn enable more sophisticated models. Beyond this, the embedding of neuroanatomical position information allows for the pooling of data across participants and even sensing modalities. This can support the creation of large, pretrained models that can be used to bootstrap model convergence and performance.

In terms of model architecture, transformer layers will likely continue to serve as the backbone of successful complex models in the short term. However, future work should explore training schemes that combine supervised and unsupervised learning, especially for their impact on online models. A paradigm whereby the majority of the learning is done in a self-supervised manner, but mistakes in the prediction can be flagged by the user, serve as ground truth, and are used in supervised learning.

An eventual speech BCI would be used continuously over months and years. This kind of hybrid modeling will likely be required to maintain model performance while compensating for the evolution of user needs and neural plasticity over time. Such a paradigm could also be used for models which slowly increase their available decoding vocabulary with extended interaction, input, and labeling from the user.

9

Curriculum Vitae

EDUCATION

Virginia Commonwealth University

2018 - present

PhD of Biomedical Engineering; 4.0 GPA

Advisor: Prof. Dean J. Krusienski

Georgia Institute of Technology

2008 - 2009

Masters of Science in Statistics; 3.33 GPA

Georgia Institute of Technology

2004 - 2008

Bachelor of Science in Applied Mathematics; 3.34 GPA

PUBLICATIONS

Lesaja, S., Stuart, M., Shih, J. J., Soroush, P. Z., Schultz, T., Manic, M., Krusienski, D. J. (2022 Submitted). Self-Supervised Learning of Neural Speech Representations from Unlabeled Intracranial Signals. *IEEE ACCESS* (2022 Submitted).

Lesaja, S., Stuart, M., Shih, J. J., Schultz, T., Manic, M., Krusienski, D. J. (2022). An Interpretable Deep Learning Model for Speech Activity Detection Using Electroencephalographic Signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30, 2783-2792.

Lesaja, S., Herff, C., Johnson, G. D., Shih, J. J., Schultz, T., Krusienski, D. J. (2019, March). Decoding lip movements during continuous speech using electroencephalography. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 522-525). IEEE.

Bibliography

- [1] Francisco Aboitiz. “A Brain for Speech. Evolutionary Continuity in Primate and Human Auditory-Vocal Processing”. In: *Frontiers in Neuroscience* 12 (2018). ISSN: 1662-453X.
- [2] Abien Fred Agarap. “Deep Learning Using Rectified Linear Units (ReLU)”. In: (2018). DOI: [10.48550/arXiv.1803.08375](https://doi.org/10.48550/arXiv.1803.08375).
- [3] Hassan Akbari et al. “Towards reconstructing intelligible speech from the human auditory cortex”. In: *Scientific reports* 9.1 (2019), pp. 1–12.
- [4] Hassan Akbari et al. “VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text”. In: (), p. 20.
- [5] Jay Alammar. *The Illustrated Transformer*. <https://jalammar.github.io/illustrated-transformer/>.
- [6] Miguel Angrick et al. “Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity”. In: *Communications Biology* 4.1 (2021), pp. 1–10.
- [7] Miguel Angrick et al. “Speech synthesis from ECoG using densely connected 3D convolutional neural networks”. In: *Journal of Neural Engineering* 3 (2019). DOI: [10.1088/1741-2552/ab0c59](https://doi.org/10.1088/1741-2552/ab0c59).

-
- [8] Gopala K. Anumanchipalli, Josh Chartier, and Edward F. Chang. “Speech synthesis from neural decoding of spoken sentences”. In: *Nature* 7753 (2019). DOI: [10.1038/s41586-019-1119-1](https://doi.org/10.1038/s41586-019-1119-1).
- [9] Anurag Arnab et al. “ViViT: A Video Vision Transformer”. In: *arXiv:2103.15691 [cs]* (2021). arXiv: [2103.15691 \[cs\]](https://arxiv.org/abs/2103.15691).
- [10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: [10.48550/arXiv.1607.06450](https://doi.org/10.48550/arXiv.1607.06450). arXiv: [1607.06450 \[cs, stat\]](https://arxiv.org/abs/1607.06450).
- [11] Arun Babu et al. “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale”. In: *arXiv:2111.09296 [cs, eess]* (2021). arXiv: [2111.09296 \[cs, eess\]](https://arxiv.org/abs/2111.09296).
- [12] Alexei Baevski, Steffen Schneider, and Michael Auli. “Vq-Wav2vec: Self-Supervised Learning of Discrete Speech Representations”. In: *arXiv:1910.05453 [cs]* (2020). arXiv: [1910.05453 \[cs\]](https://arxiv.org/abs/1910.05453).
- [13] Alexei Baevski et al. “Unsupervised Speech Recognition”. In: *arXiv:2105.11084 [cs, eess]* (2021). arXiv: [2105.11084 \[cs, eess\]](https://arxiv.org/abs/2105.11084).
- [14] Alexei Baevski et al. “Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”. In: *arXiv:2006.11477 [cs, eess]* (2020). arXiv: [2006.11477 \[cs, eess\]](https://arxiv.org/abs/2006.11477).
- [15] Tonio Ball et al. “Signal quality of simultaneously recorded invasive and non-invasive EEG”. In: *Neuroimage* 46.3 (2009), pp. 708–716.
- [16] J Bancaud. “Apport de l’exploration fonctionnelle par voie stéréotaxique à la chirurgie de l’épilepsie”. In: *Neurochirurgie* 5.1 (1959), pp. 55–112.

- [17] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is Space-Time Attention All You Need for Video Understanding?" In: *arXiv:2102.05095 [cs]* (2021). arXiv: [2102.05095 \[cs\]](https://arxiv.org/abs/2102.05095).
- [18] D. Boatman et al. "Language Recovery after Left Hemispherectomy in Children with Late-Onset Seizures". In: *Annals of Neurology* 46.4 (1999), pp. 579–586. ISSN: 0364-5134. DOI: [10 . 1002 / 1531 - 8249\(199910 \) 46 : 4<579::aid-ana5>3.0.co;2-k](https://doi.org/10.1002/1531-8249(199910)46:4<579::aid-ana5>3.0.co;2-k).
- [19] Kristofer E. Bouchard et al. "Functional organization of human sensorimotor cortex for speech articulation". In: *Nature* 7441 (2013). DOI: [10 . 1038/nature11911](https://doi.org/10.1038/nature11911).
- [20] Paul Broca et al. "Perte de la parole, ramollissement chronique et destruction partielle du lobe antérieur gauche du cerveau". In: *Bull Soc Anthropol* 2.1 (1861), pp. 235–238.
- [21] Tom B. Brown et al. "Language Models Are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (2020). arXiv: [2005.14165 \[cs\]](https://arxiv.org/abs/2005.14165).
- [22] Jonathan S Brumberg et al. "Classification of intended phoneme production from chronic intracortical microelectrode recordings in speech motor cortex". In: *Frontiers in neuroscience* 5 (2011), p. 65.
- [23] Daniel Carey et al. "Functional and quantitative MRI mapping of somatomotor representations of human supralaryngeal vocal tract". In: *Cerebral Cortex* 27.1 (2017), pp. 265–278.
- [24] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *arXiv:2005.12872 [cs]* (2020). arXiv: [2005.12872 \[cs\]](https://arxiv.org/abs/2005.12872).

- [25] Jonathan Chabout et al. "A Foxp2 Mutation Implicated in Human Speech Deficits Alters Sequencing of Ultrasonic Vocalizations in Adult Male Mice". In: *Frontiers in Behavioral Neuroscience* 10 (2016), p. 197. ISSN: 1662-5153. DOI: [10.3389/fnbeh.2016.00197](https://doi.org/10.3389/fnbeh.2016.00197).
- [26] Shreya Chakrabarti et al. "Progress in speech decoding from the electrocorticogram". In: *Biomedical Engineering Letters* 1 (2015). DOI: [10.1007/s13534-015-0175-1](https://doi.org/10.1007/s13534-015-0175-1).
- [27] Josh Chartier et al. "Encoding of Articulatory Kinematic Trajectories in Human Speech Sensorimotor Cortex". In: *Neuron* 5 (2018). DOI: [10.1016/j.neuron.2018.04.031](https://doi.org/10.1016/j.neuron.2018.04.031).
- [28] Francine Chassoux et al. "Planning and management of SEEG". In: *Neurophysiologie Clinique* 48.1 (2018), pp. 25–37.
- [29] Ujwal Chaudhary, Niels Birbaumer, and Ander Ramos-Murguialday. "Brain–Computer Interfaces for Communication and Rehabilitation". In: *Nature Reviews Neurology* 12.9 (2016), pp. 513–525. ISSN: 1759-4766. DOI: [10.1038/nrneurol.2016.113](https://doi.org/10.1038/nrneurol.2016.113).
- [30] Sihong Chen, Kai Ma, and Yefeng Zheng. *Med3D: Transfer Learning for 3D Medical Image Analysis*. 2019. DOI: [10.48550/arXiv.1904.00625](https://doi.org/10.48550/arXiv.1904.00625). arXiv: [1904.00625](https://arxiv.org/abs/1904.00625) [cs].
- [31] Ting Chen et al. "Big Self-Supervised Models Are Strong Semi-Supervised Learners". In: *arXiv:2006.10029 [cs, stat]* (2020). arXiv: [2006.10029](https://arxiv.org/abs/2006.10029) [cs, stat].
- [32] Yu-An Chung et al. "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training". In: *arXiv:2108.06209 [cs, eess]* (2021). arXiv: [2108.06209](https://arxiv.org/abs/2108.06209) [cs, eess].

- [33] Béla Clemens et al. "Quantitative EEG Effects of Carbamazepine, Oxcarbazepine, Valproate, Lamotrigine, and Possible Clinical Relevance of the Findings". In: *Epilepsy Research* 70.2 (2006), pp. 190–199. ISSN: 0920-1211. DOI: [10.1016/j.eplepsyres.2006.05.003](https://doi.org/10.1016/j.eplepsyres.2006.05.003).
- [34] Chris Code. "Can the right hemisphere speak?" In: *Brain and Language* 57.1 (1997), pp. 38–59.
- [35] Louis Collins. "3D Model-based segmentation of individual brain structures from magnetic resonance imaging data". In: (1994).
- [36] Arden Dertat. *Applied Deep Learning - Part 4: Convolutional Neural Networks*. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>. 2017.
- [37] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (2019). arXiv: [1810.04805 \[cs\]](https://arxiv.org/abs/1810.04805).
- [38] Anthony Steven Dick et al. "Co-Speech Gestures Influence Neural Activity in Brain Regions Associated with Processing Semantic Information". In: *Human Brain Mapping* 30.11 (2009), pp. 3509–3526. ISSN: 1097-0193. DOI: [10.1002/hbm.20774](https://doi.org/10.1002/hbm.20774).
- [39] Synho Do, Kyoung Song, and Joo Chung. "Basics of Deep Learning: A Radiologist's Guide to Understanding Published Radiology Articles on Deep Learning". In: *Korean journal of radiology* 21 (2020), pp. 33–41. DOI: [10.3348/kjr.2019.0312](https://doi.org/10.3348/kjr.2019.0312).
- [40] Alexey Dosovitskiy et al. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv:2010.11929 [cs]* (2021). arXiv: [2010.11929 \[cs\]](https://arxiv.org/abs/2010.11929).

- [41] A.C. Evans et al. "3D Statistical Neuroanatomical Models from 305 MRI Volumes". In: *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*. 1993, 1813–1817 vol.3. DOI: [10.1109/NSSMIC.1993.373602](https://doi.org/10.1109/NSSMIC.1993.373602).
- [42] Hehe Fan, Yi Yang, and Mohan Kankanhalli. "Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, 2021, pp. 14199–14208. ISBN: 978-1-66544-509-2. DOI: [10.1109/CVPR46437.2021.01398](https://doi.org/10.1109/CVPR46437.2021.01398).
- [43] Bruce Fischl. "FreeSurfer". In: *NeuroImage* 62.2 (2012), pp. 774–781. ISSN: 1053-8119. DOI: [10.1016/j.neuroimage.2012.01.021](https://doi.org/10.1016/j.neuroimage.2012.01.021).
- [44] Kosuke Fukumori et al. "Epileptic Spike Detection Using Neural Networks With Linear-Phase Convolutions". In: *IEEE Journal of Biomedical and Health Informatics* 26.3 (2022), pp. 1045–1056. ISSN: 2168-2194, 2168-2208. DOI: [10.1109/JBHI.2021.3102247](https://doi.org/10.1109/JBHI.2021.3102247).
- [45] Kunihiko Fukushima and Sei Miyake. "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position". In: *Pattern Recognition* 15.6 (1982), pp. 455–469. ISSN: 0031-3203. DOI: [10.1016/0031-3203\(82\)90024-3](https://doi.org/10.1016/0031-3203(82)90024-3).
- [46] Hanna S. Gauvin and Robert J. Hartsuiker. "Towards a New Model of Verbal Monitoring". In: *Journal of Cognition* 3.1 (2020), p. 17. ISSN: 2514-4820. DOI: [10.5334/joc.81](https://doi.org/10.5334/joc.81).
- [47] Sid Gilman, Sarah Winans Newman, and John Tinkham Manter. *Manter and Gatz's essentials of clinical neuroanatomy and neurophysiology*. FA Davis Company, 1996.

- [48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [49] Alexandre Gramfort et al. “MEG and EEG Data Analysis with MNE-Python”. In: *Frontiers in Neuroscience* 7.267 (2013), pp. 1–13. DOI: [10.3389/fnins.2013.00267](https://doi.org/10.3389/fnins.2013.00267).
- [50] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [51] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.
- [52] Lili He et al. “A Multi-Task, Multi-Stage Deep Transfer Learning Model for Early Prediction of Neurodevelopment in Very Preterm Infants”. In: *Scientific Reports* 10.1 (2020), p. 15072. ISSN: 2045-2322. DOI: [10.1038/s41598-020-71914-x](https://doi.org/10.1038/s41598-020-71914-x).
- [53] C. Henley. *Foundations of Neuroscience*. Open Textbook Library. Michigan State University, 2021.
- [54] Christian Herff, Dean J. Krusienski, and Pieter Kubben. “The Potential of Stereotactic-EEG for Brain-Computer Interfaces: Current Progress and Future Directions”. In: *Frontiers in Neuroscience* (2020). DOI: [10.3389/fnins.2020.00123](https://doi.org/10.3389/fnins.2020.00123).
- [55] Christian Herff et al. “Brain-to-text: decoding spoken phrases from phone representations in the brain”. In: *Frontiers in Neuroscience* (2015). DOI: [10.3389/fnins.2015.00217](https://doi.org/10.3389/fnins.2015.00217).

- [56] Christian Herff et al. “Generating Natural, Intelligible Speech From Brain Activity in Motor, Premotor, and Inferior Frontal Cortices”. In: *Frontiers in Neuroscience* (2019). DOI: [10.3389/fnins.2019.01267](https://doi.org/10.3389/fnins.2019.01267).
- [57] Christian Herff et al. “Mental Workload during N-Back Task—Quantified in the Prefrontal Cortex Using fNIRS”. In: *Frontiers in Human Neuroscience* 7 (2014). ISSN: 1662-5161.
- [58] Christian Herff et al. “Towards Direct Speech Synthesis from ECoG: A Pilot Study”. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Orlando, FL, USA: IEEE, 2016, pp. 1540–1543. ISBN: 978-1-4577-0220-4. DOI: [10.1109/EMBC.2016.7591004](https://doi.org/10.1109/EMBC.2016.7591004).
- [59] Gregory Hickok. “Chapter 4 - The Dual Stream Model of Speech and Language Processing”. In: *Handbook of Clinical Neurology*. Ed. by Argye Elizabeth Hillis and Julius Fridriksson. Vol. 185. Aphasia. Elsevier, 2022, pp. 57–69. DOI: [10.1016/B978-0-12-823384-9.00003-7](https://doi.org/10.1016/B978-0-12-823384-9.00003-7).
- [60] Gregory Hickok and David Poeppel. “The cortical organization of speech processing”. In: *Nature reviews neuroscience* 8.5 (2007), pp. 393–402.
- [61] Geoffrey Hinton and Terrence J Sejnowski. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.
- [62] Leigh R. Hochberg et al. “Reach and Grasp by People with Tetraplegia Using a Neurally Controlled Robotic Arm”. In: *Nature* 485.7398 (2012), pp. 372–375. ISSN: 1476-4687. DOI: [10.1038/nature11076](https://doi.org/10.1038/nature11076).
- [63] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. DOI: [10.48550/arXiv.2203.15556](https://doi.org/10.48550/arXiv.2203.15556). arXiv: 2203.15556 [cs].

- [64] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 0893-6080. DOI: [10.1016/0893-6080\(91\)90009-T](https://doi.org/10.1016/0893-6080(91)90009-T).
- [65] Guy Hotson et al. "Individual Finger Control of a Modular Prosthetic Limb Using High-Density Electrocorticography in a Human Subject". In: *Journal of Neural Engineering* 13.2 (2016), p. 026017. ISSN: 1741-2552. DOI: [10.1088/1741-2560/13/2/026017](https://doi.org/10.1088/1741-2560/13/2/026017).
- [66] Arthur S House et al. "Psychoacoustic speech tests: A modified rhyme test". In: *The Journal of the Acoustical Society of America* 35.11 (1963), pp. 1899–1899.
- [67] Wei-Ning Hsu et al. "Hubert: How Much Can a Bad Teacher Benefit ASR Pre-Training?" In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 6533–6537. DOI: [10.1109/ICASSP39728.2021.9414460](https://doi.org/10.1109/ICASSP39728.2021.9414460).
- [68] Wei-Ning Hsu et al. "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". In: *arXiv:2106.07447 [cs, eess]* (2021). arXiv: [2106.07447 \[cs, eess\]](https://arxiv.org/abs/2106.07447).
- [69] Amy L. Hubbard et al. "Giving Speech a Hand: Gesture Modulates Activity in Auditory Cortex during Speech Perception". In: *Human Brain Mapping* 30.3 (2009), pp. 1028–1037. ISSN: 1097-0193. DOI: [10.1002/hbm.20565](https://doi.org/10.1002/hbm.20565).
- [70] D. H. Hubel and T. N. Wiesel. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex". In: *The Journal of Physiology* 160.1 (1962), pp. 106–154.2. ISSN: 0022-3751.

- [71] Jane E. Huggins et al. “Workshops of the Sixth International Brain–Computer Interface Meeting: brain–computer interfaces past, present, and future”. In: *Brain-Computer Interfaces 1-2* (2017). DOI: [10.1080/2326263X.2016.1275488](https://doi.org/10.1080/2326263X.2016.1275488).
- [72] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [73] Peter Indefrey and Willem JM Levelt. “The spatial and temporal signatures of word production components”. In: *Cognition* 92.1-2 (2004), pp. 101–144.
- [74] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. en. In: (Feb. 2015). URL: <https://arxiv.org/abs/1502.03167v3> (visited on 09/06/2020).
- [75] Eric Jang, Shixiang Gu, and Ben Poole. *Categorical Reparameterization with Gumbel-Softmax*. 2017. DOI: [10.48550/arXiv.1611.01144](https://doi.org/10.48550/arXiv.1611.01144). arXiv: [1611.01144](https://arxiv.org/abs/1611.01144) [cs, stat].
- [76] H Jégou, M Douze, and C Schmid. “Product Quantization for Nearest Neighbor Search”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1 (2011), pp. 117–128. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2010.57](https://doi.org/10.1109/TPAMI.2010.57).
- [77] Vasileios G Kanas et al. “Joint spatial-spectral feature space clustering for speech activity detection from ECoG signals”. In: *IEEE Transactions on Biomedical Engineering* 61.4 (2014), pp. 1241–1250.

- [78] Vasileios G Kanas et al. "Real-time voice activity detection for ECoG-based speech brain machine interfaces". In: *2014 19th International Conference on Digital Signal Processing*. IEEE. 2014, pp. 862–865.
- [79] Andrej Karpathy. "Connecting images and natural language". PhD thesis. Stanford University, 2016.
- [80] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).
- [81] S. Knecht et al. "Language Lateralization in Healthy Right-Handers". In: *Brain* 123.1 (2000), pp. 74–81. ISSN: 0006-8950. DOI: [10.1093/brain/123.1.74](https://doi.org/10.1093/brain/123.1.74).
- [82] Jonas Kohler et al. "Synthesizing Speech from Intracranial Depth Electrodes Using an Encoder-Decoder Framework". In: *arXiv:2111.01457 [cs]* (2021). arXiv: [2111.01457 \[cs\]](https://arxiv.org/abs/2111.01457).
- [83] Shuji Komeiji et al. "Transformer-Based Estimation of Spoken Sentences Using Electrocorticography". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 1311–1315. DOI: [10.1109/ICASSP43922.2022.9747443](https://doi.org/10.1109/ICASSP43922.2022.9747443).
- [84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90. ISSN: 0001-0782. DOI: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [85] Max O. Krucoff et al. "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation". In: *Frontiers in Neuroscience* 10 (2016), p. 584. ISSN: 1662-4548. DOI: [10.3389/fnins.2016.00584](https://doi.org/10.3389/fnins.2016.00584).

- [86] Dean J Krusienski et al. "Toward enhanced P300 speller performance". In: *Journal of neuroscience methods* 167.1 (2008), pp. 15–21.
- [87] Jan Kubanek and Gerwin Schalk. "NeuralAct: a tool to visualize electrocortical (ECoG) activity on a three-dimensional model of the cortex". In: *Neuroinformatics* 13.2 (2015), pp. 167–174.
- [88] Peter Langland-Hassan and Agustín Vicente. *Inner speech: New voices*. Oxford University Press, USA, 2018.
- [89] Vernon J Lawhern et al. "EEGNet: A Compact Convolutional Neural Network for EEG-based Brain–Computer Interfaces". In: *Journal of Neural Engineering* 15.5 (2018), p. 056013. ISSN: 1741-2560, 1741-2552. DOI: [10.1088/1741-2552/aace8c](https://doi.org/10.1088/1741-2552/aace8c).
- [90] Y. Lecun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 1558-2256. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [91] Eric Leuthardt et al. "Temporal Evolution of Gamma Activity in Human Cortex during an Overt and Covert Word Repetition Task". In: *Frontiers in Human Neuroscience* 6 (2012). ISSN: 1662-5161.
- [92] Willem J. M. Levelt, Ardi Roelofs, and Antje S. Meyer. "A Theory of Lexical Access in Speech Production". In: *Behavioral and Brain Sciences* 22.1 (1999), pp. 1–38. ISSN: 1469-1825, 0140-525X. DOI: [10.1017/S0140525X99001776](https://doi.org/10.1017/S0140525X99001776).
- [93] Guangye Li et al. "Detection of human white matter activation and evaluation of its function in movement decoding using stereo-electroencephalography (sEEG)". In: *Journal of Neural Engineering* 18.4 (2021), p. 0460c6.

- [94] Alvin M. Liberman and Doug H. Whalen. "On the Relation of Speech to Language". In: *Trends in Cognitive Sciences* 4.5 (2000), pp. 187–196. ISSN: 1364-6613. DOI: [10.1016/S1364-6613\(00\)01471-6](https://doi.org/10.1016/S1364-6613(00)01471-6).
- [95] Ludwig Lichtheim. "On aphasia". In: *Brain* 7 (1885), pp. 433–484.
- [96] Patricia Limousin and Tom Foltynie. "Long-Term Outcomes of Deep Brain Stimulation in Parkinson Disease". In: *Nature Reviews Neurology* 15.4 (2019), pp. 234–242. ISSN: 1759-4766. DOI: [10.1038/s41582-019-0145-9](https://doi.org/10.1038/s41582-019-0145-9).
- [97] Lars E van der Loo et al. "Methodology, outcome, safety and in vivo accuracy in traditional frame-based stereoelectroencephalography". In: *Acta neurochirurgica* 159.9 (2017), pp. 1733–1746.
- [98] Fabien Lotte et al. "Electrocorticographic representations of segmental features in continuous speech". In: *Frontiers in Human Neuroscience* (2015). DOI: [10.3389/fnhum.2015.00097](https://doi.org/10.3389/fnhum.2015.00097).
- [99] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).
- [100] Joseph G. Makin, David A. Moses, and Edward F. Chang. "Machine translation of cortical activity to text with an encoder–decoder framework". In: *Nature Neuroscience* 4 (2020). DOI: [10.1038/s41593-020-0608-8](https://doi.org/10.1038/s41593-020-0608-8).
- [101] Stephanie Martin et al. "Decoding Inner Speech Using Electrocorticography: Progress and Challenges Toward a Speech Prosthesis". In: *Frontiers in Neuroscience* (2018). DOI: [10.3389/fnins.2018.00422](https://doi.org/10.3389/fnins.2018.00422).
- [102] Stéphanie Martin et al. "Decoding Spectrotemporal Features of Overt and Covert Speech from the Human Cortex". In: *Frontiers in Neuroengineering* 7 (2014), p. 14. ISSN: 1662-6443. DOI: [10.3389/fneng.2014.00014](https://doi.org/10.3389/fneng.2014.00014).

- [103] Stephanie Martin et al. “Word pair classification during imagined speech using direct brain recordings”. In: *Scientific Reports* 1 (2016). DOI: [10.1038/srep25803](https://doi.org/10.1038/srep25803).
- [104] *MATLAB version 9.3.0.713579 (R2017b)*. The Mathworks, Inc. Natick, Massachusetts, 2017.
- [105] Warren S. McCulloch and Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133. ISSN: 1522-9602. DOI: [10.1007/BF02478259](https://doi.org/10.1007/BF02478259).
- [106] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).
- [107] M-Marsel Mesulam. “Large-Scale Neurocognitive Networks and Distributed Processing for Attention, Language, and Memory”. In: *Annals of Neurology* 28.5 (1990), pp. 597–613. ISSN: 1531-8249. DOI: [10.1002/ana.410280502](https://doi.org/10.1002/ana.410280502).
- [108] M Ardussi Mines, Barbara F Hanson, and June E Shoup. “Frequency of occurrence of phonemes in conversational English”. In: *Language and speech* 21.3 (1978), pp. 221–241.
- [109] David A. Moses et al. “Neural Speech Recognition: Continuous Phoneme Decoding Using Spatiotemporal Representations of Human Cortical Activity”. In: *Journal of Neural Engineering* 13.5 (2016), p. 056004. ISSN: 1741-2552. DOI: [10.1088/1741-2560/13/5/056004](https://doi.org/10.1088/1741-2560/13/5/056004).
- [110] David A. Moses et al. “Neuroprosthesis for Decoding Speech in a Paralyzed Person with Anarthria”. In: *New England Journal of Medicine* 3 (2021). DOI: [10.1056/NEJMoa2027540](https://doi.org/10.1056/NEJMoa2027540).

- [111] David A. Moses et al. "Real-time decoding of question-and-answer speech dialogue using human cortical activity". In: *Nature Communications* 1 (2019). DOI: [10.1038/s41467-019-10994-4](https://doi.org/10.1038/s41467-019-10994-4).
- [112] Emily M. Mugler et al. "Direct classification of all American English phonemes using signals from functional speech motor cortex". In: *Journal of Neural Engineering* 3 (2014). DOI: [10.1088/1741-2560/11/3/035015](https://doi.org/10.1088/1741-2560/11/3/035015).
- [113] Ewan S. Nurse et al. "Consistency of Long-Term Subdural Electrocor-ticography in Humans". In: *IEEE Transactions on Biomedical Engineering* 65.2 (2018), pp. 344–352. ISSN: 1558-2531. DOI: [10.1109/TBME.2017.2768442](https://doi.org/10.1109/TBME.2017.2768442).
- [114] Seiji Ogawa et al. "Brain magnetic resonance imaging with contrast de-pendent on blood oxygenation." In: *proceedings of the National Academy of Sciences* 87.24 (1990), pp. 9868–9872.
- [115] George Ojemann et al. "Cortical Language Localization in Left, Domi-nant Hemisphere: An Electrical Stimulation Mapping Investigation in 117 Patients". In: *Journal of Neurosurgery* 71.3 (1989), pp. 316–326. DOI: [10.3171/jns.1989.71.3.0316](https://doi.org/10.3171/jns.1989.71.3.0316).
- [116] Brian N Pasley et al. "Reconstructing speech from human auditory cor-tex". In: *PLoS biology* 10.1 (2012), e1001251.
- [117] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Sys-tems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

- [118] Mark Patkowski. “Laterality effects in multilinguals during speech production under the concurrent task paradigm: Another test of the age of acquisition hypothesis”. In: (2003).
- [119] Steven M. Peterson et al. “Generalized Neural Decoders for Transfer Learning across Participants and Recording Modalities”. In: *Journal of Neural Engineering* 18.2 (2021), p. 026014. ISSN: 1741-2552. DOI: [10.1088/1741-2552/abda0b](https://doi.org/10.1088/1741-2552/abda0b).
- [120] David Poeppel. “The Neuroanatomic and Neurophysiological Infrastructure for Speech and Language”. In: *Current Opinion in Neurobiology*. SI: Communication and Language 28 (2014), pp. 142–149. ISSN: 0959-4388. DOI: [10.1016/j.conb.2014.07.005](https://doi.org/10.1016/j.conb.2014.07.005).
- [121] Timothée Proix et al. “Imagined Speech Can Be Decoded from Low- and Cross-Frequency Intracranial EEG Features”. In: *Nature Communications* 13.1 (2022), p. 48. ISSN: 2041-1723. DOI: [10.1038/s41467-021-27725-3](https://doi.org/10.1038/s41467-021-27725-3).
- [122] Qinwan Rabbani, Griffin Milsap, and Nathan E. Crone. “The Potential for a Speech Brain–Computer Interface Using Chronic Electrocorticography”. In: *Neurotherapeutics* 1 (2019). DOI: [10.1007/s13311-018-00692-2](https://doi.org/10.1007/s13311-018-00692-2).
- [123] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. DOI: [10.48550/arXiv.2103.00020](https://doi.org/10.48550/arXiv.2103.00020). arXiv: [2103.00020](https://arxiv.org/abs/2103.00020) [cs].
- [124] Jack W. Rae et al. *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. 2022. DOI: [10.48550/arXiv.2112.11446](https://doi.org/10.48550/arXiv.2112.11446). arXiv: [2112.11446](https://arxiv.org/abs/2112.11446) [cs].

- [125] Maithra Raghu et al. *Transfusion: Understanding Transfer Learning for Medical Imaging*. 2019. DOI: [10.48550/arXiv.1902.07208](https://doi.org/10.48550/arXiv.1902.07208). arXiv: [1902.07208](https://arxiv.org/abs/1902.07208) [cs, stat].
- [126] Aditya Ramesh et al. *Zero-Shot Text-to-Image Generation*. 2021. DOI: [10.48550/arXiv.2102.12092](https://doi.org/10.48550/arXiv.2102.12092). arXiv: [2102.12092](https://arxiv.org/abs/2102.12092) [cs].
- [127] Josef P. Rauschecker and Sophie K. Scott. “Maps and Streams in the Auditory Cortex: Nonhuman Primates Illuminate Human Speech Processing”. In: *Nature Neuroscience* 12.6 (2009), pp. 718–724. ISSN: 1546-1726. DOI: [10.1038/nn.2331](https://doi.org/10.1038/nn.2331).
- [128] Mirco Ravanelli and Yoshua Bengio. “Speaker recognition from raw waveform with sincnet”. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 1021–1028.
- [129] Mirco Ravanelli, Titouan Parcollet, and Yoshua Bengio. “The Pytorchkaldi Speech Recognition Toolkit”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019. DOI: [10.1109/ICASSP.2019.8683713](https://doi.org/10.1109/ICASSP.2019.8683713).
- [130] Mirco Ravanelli et al. “Multi-Task Self-Supervised Learning for Robust Speech Recognition”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 6989–6993. DOI: [10.1109/ICASSP40776.2020.9053569](https://doi.org/10.1109/ICASSP40776.2020.9053569).
- [131] Martin Reuter et al. “Within-Subject Template Estimation for Unbiased Longitudinal Image Analysis”. In: *NeuroImage* 61.4 (2012), pp. 1402–1418. ISSN: 1053-8119. DOI: [10.1016/j.neuroimage.2012.02.084](https://doi.org/10.1016/j.neuroimage.2012.02.084).
- [132] Andrew Y Revell et al. “White matter signals reflect information transmission between brain regions during seizures”. In: *BioRxiv* (2021).

- [133] Ardi Roelofs. "Spoken Word Planning, Comprehending, and Self-Monitoring: Evaluation of WEAVER++". In: *Phonological Encoding and Monitoring in Normal and Pathological Speech*. New York, NY, US: Psychology Press, 2005, pp. 42–63. ISBN: 978-1-84169-262-3.
- [134] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". In: *Psychological Review* 65 (1958), pp. 386–408. ISSN: 1939-1471. DOI: [10.1037/h0042519](https://doi.org/10.1037/h0042519).
- [135] EH Rothauser. "IEEE recommended practice for speech quality measurements". In: *IEEE Trans. on Audio and Electroacoustics* 17 (1969), pp. 225–246.
- [136] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 1573-1405. DOI: [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y).
- [137] G. Schalk et al. "BCI2000: a general-purpose brain-computer interface (BCI) system". In: *IEEE Transactions on Biomedical Engineering* 6 (2004). DOI: [10.1109/TBME.2004.827072](https://doi.org/10.1109/TBME.2004.827072).
- [138] G. Schalk et al. "Two-Dimensional Movement Control Using Electrocor-ticographic Signals in Humans". In: *Journal of Neural Engineering* 5.1 (2008), pp. 75–84. ISSN: 1741-2552. DOI: [10.1088/1741-2560/5/1/008](https://doi.org/10.1088/1741-2560/5/1/008).
- [139] Gerwin Schalk and Eric C Leuthardt. "Brain-computer interfaces using electrocorticographic signals". In: *IEEE reviews in biomedical engineering* 4 (2011), pp. 140–154.
- [140] R. Schirrmeister et al. "Deep learning with convolutional neural networks for decoding and visualization of EEG pathology". In: *2017 IEEE Signal*

- Processing in Medicine and Biology Symposium (SPMB)*. IEEE, 2017. DOI: [10.1109/SPMB.2017.8257015](https://doi.org/10.1109/SPMB.2017.8257015).
- [141] Shayne Shaw et al. *Teacher-Student Chain for Efficient Semi-Supervised Histology Image Classification*. 2020. DOI: [10.48550/arXiv.2003.08797](https://doi.org/10.48550/arXiv.2003.08797). arXiv: [2003.08797](https://arxiv.org/abs/2003.08797) [cs, eess, stat].
- [142] Jerry J. Shih, Dean J. Krusienski, and Jonathan R. Wolpaw. "Brain-Computer Interfaces in Medicine". In: *Mayo Clinic Proceedings* 87.3 (2012), pp. 268–279. ISSN: 0025-6196. DOI: [10.1016/j.mayocp.2011.12.008](https://doi.org/10.1016/j.mayocp.2011.12.008).
- [143] Kåre Sjölander and Jonas Beskow. *Wavesurfer - An Open Source Speech Tool*. 2000.
- [144] Marc W. Slutzky and Robert D. Flint. "Physiological properties of brain-machine interface input signals". In: *Journal of Neurophysiology* 2 (2017). DOI: [10.1152/jn.00070.2017](https://doi.org/10.1152/jn.00070.2017).
- [145] Shaden Smith et al. *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*. 2022. DOI: [10.48550/arXiv.2201.11990](https://doi.org/10.48550/arXiv.2201.11990). arXiv: [2201.11990](https://arxiv.org/abs/2201.11990) [cs].
- [146] PZ Soroush et al. "Speech Activity Detection from Stereotactic EEG". In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2021, pp. 3402–3407.
- [147] Morgan Stuart and Milos Manic. "Deep Learning Shared Bandpass Filters for Resource-Constrained Human Activity Recognition". In: *IEEE Access* 9 (2021), pp. 39089–39097.
- [148] Kayt Sukel. *Neuroanatomy: The Basics*. <https://www.dana.org/article/neuroanatomy-the-basics/>. 2011.

- [149] J. P. Szaflarski et al. "Language Lateralization in Left-Handed and Ambidextrous People: fMRI Data". In: *Neurology* 59.2 (2002), pp. 238–244. ISSN: 0028-3878, 1526-632X. DOI: [10.1212/WNL.59.2.238](https://doi.org/10.1212/WNL.59.2.238).
- [150] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134). URL: <https://doi.org/10.5281/zenodo.3509134>.
- [151] Jonathan Tompson et al. "Efficient Object Localization Using Convolutional Networks". In: *CoRR abs/1411.4280* (2014). arXiv: [1411.4280](https://arxiv.org/abs/1411.4280). URL: <http://arxiv.org/abs/1411.4280>.
- [152] Christoph Tremmel et al. "Estimating Cognitive Workload in an Interactive Virtual Reality Environment Using EEG". In: *Frontiers in Human Neuroscience* 13 (2019). ISSN: 1662-5161.
- [153] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762 [cs]* (2017). arXiv: [1706.03762 \[cs\]](https://arxiv.org/abs/1706.03762).
- [154] Maxime Verwoert et al. "Dataset of Speech Production in Intracranial Electroencephalography". In: *Scientific Data* 9.1 (2022), p. 434. ISSN: 2052-4463. DOI: [10.1038/s41597-022-01542-9](https://doi.org/10.1038/s41597-022-01542-9).
- [155] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [156] Ksenia Volkova et al. "Decoding Movement From Electrocorticographic Activity: A Review". In: *Frontiers in Neuroinformatics* 13 (2019). ISSN: 1662-5196.

- [157] Jonas Wacker, Marcelo Ladeira, and José Eduardo Vaz Nascimento. *Transfer Learning for Brain Tumor Segmentation*. 2020. DOI: [10.48550/arXiv.1912.12452](https://doi.org/10.48550/arXiv.1912.12452). arXiv: [1912.12452](https://arxiv.org/abs/1912.12452) [cs, eess].
- [158] Alexander Waibel et al. "Phoneme recognition using time-delay neural networks". In: *IEEE transactions on acoustics, speech, and signal processing* 37.3 (1989), pp. 328–339.
- [159] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021). URL: <https://doi.org/10.21105/joss.03021>.
- [160] Carl Wernicke. "The aphasic symptom-complex: a psychological study on an anatomical basis". In: *Archives of Neurology* 22.3 (1970), pp. 280–282.
- [161] Roel M. Willems, Aslı Özyürek, and Peter Hagoort. "When Language Meets Action: The Neural Integration of Gesture and Speech". In: *Cerebral Cortex* 17.10 (2007), pp. 2322–2333. ISSN: 1047-3211. DOI: [10.1093/cercor/bhl141](https://doi.org/10.1093/cercor/bhl141).
- [162] Paul Wimmer, Jens Mehnert, and Alexandru Condurache. "FreezeNet: Full Performance by Reduced Storage Costs". In: vol. 12627. 2021, pp. 685–701. DOI: [10.1007/978-3-030-69544-6_41](https://doi.org/10.1007/978-3-030-69544-6_41). arXiv: [2011.14087](https://arxiv.org/abs/2011.14087) [cs].
- [163] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal Component Analysis". In: *Chemometrics and Intelligent Laboratory Systems*. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists 2.1 (1987), pp. 37–52. ISSN: 0169-7439. DOI: [10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).

-
- [164] Jonathan R Wolpaw et al. “Brain–computer interfaces for communication and control”. In: *Clinical Neurophysiology* 6 (2002). DOI: [10.1016/S1388-2457\(02\)00057-3](https://doi.org/10.1016/S1388-2457(02)00057-3).
- [165] Lei Xiao et al. “Expression of FoxP2 in the Basal Ganglia Regulates Vocal Motor Sequences in the Adult Songbird”. In: *Nature Communications* 12.1 (2021), p. 2617. ISSN: 2041-1723. DOI: [10.1038/s41467-021-22918-2](https://doi.org/10.1038/s41467-021-22918-2).
- [166] Jiang Xu et al. “Symbolic Gestures and Spoken Language Are Processed by a Common Neural System”. In: *Proceedings of the National Academy of Sciences* 106.49 (2009), pp. 20664–20669. DOI: [10.1073/pnas.0909197106](https://doi.org/10.1073/pnas.0909197106).
- [167] Jason Yosinski et al. *How Transferable Are Features in Deep Neural Networks?* 2014. arXiv: [1411.1792 \[cs\]](https://arxiv.org/abs/1411.1792).
- [168] Yang Zhang and Yue Wang. “Neural Plasticity in Speech Acquisition and Learning”. In: *Bilingualism: Language and Cognition* 10.2 (2007), pp. 147–160. ISSN: 1469-1841, 1366-7289. DOI: [10.1017/S1366728907002908](https://doi.org/10.1017/S1366728907002908).