Neural Speech Decoding and Understanding Leveraging Deep Learning and Speech Synthesis

---

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Electrical and Computer Engineering)

at the

NEW YORK UNIVERSITY

TANDON SCHOOL OF ENGINEERING

by

Xupeng Chen

May 2025

Neural Speech Decoding and Understanding Leveraging Deep Learning and Speech Synthesis

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

NEW YORK UNIVERSITY

TANDON SCHOOL OF ENGINEERING

by

Xupeng Chen

May 2025

Approved:

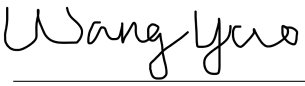
Department Chair Signature

May 8, 2025

Date

Approved by the Guidance Committee:

Major: Electrical and Computer Engineering



Yao Wang
Professor of
NYU Tandon School of Engineering

5/8/2025
Date

Adeen Flinker
Associate Professor of
NYU Grossman School of Medicine
NYU Tandon School of Engineering

5/8/2025
Date

Anna Choromanska
Associate Professor of
NYU Tandon School of Engineering
NYU Center for Data Science

05/04/2025
Date

Vita

Xupeng Chen received his Bachelor of Science in Life Science, with a minor in Statistics, from Tsinghua University in June 2019, where he was a member of the XueTang program for cultivating leading research talent. He joined the Tandon School of Engineering at New York University in September 2019 as a Ph.D. candidate in Electrical Engineering under the supervision of Prof. Yao Wang and Prof. Adeen Flinker. His doctoral work focuses on designing deep learning–based frameworks for neural speech decoding from electrocorticographic (ECoG), stereo-electroencephalographic (sEEG), and surface electromyographic (sEMG) signals. He has been honored with the 2025 Dr. Li Annual ECE Publication Award and the 2024 Dante Youla Award for Graduate Research Excellence in Electrical Engineering at NYU.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Yao Wang, for her unwavering support and guidance throughout my Ph.D. years. Her encouragement, expertise, and insight have been invaluable to every aspect of my research. She taught me to approach scientific problems with both attention to detail and a deep curiosity for their fundamental nature. Her dedication and genuine interest in every aspect of my work shaped the way I think about research and made me feel truly seen and supported. Her care extended beyond research—she guided me with kindness and thoughtfulness in my personal growth as well. I am especially thankful for her thoughtful advice, which helped me grow both as a researcher and as a person.

I am also sincerely grateful to Prof. Adeen Flinker, my co-advisor, for his generous support, mentorship, and immense patience. He explained countless concepts and principles, answered my questions with clarity, and guided me through every step of writing and revising my papers. His passionate involvement and detailed feedback were instrumental in shaping the direction and quality of this research.

I also thank Prof. Anna Choromanska for her valuable feedback and guidance.

During my Ph.D. journey, I had the privilege of working with an incredible group of colleagues and friends in the Video Lab and Flinker's Lab, including Ran Wang, Amirhossein Khalilian-Gourtani, Junbo Chen, Chenqian Le, Nikasadat Emami, Tianyu He, Jianghao Qian, Wenqi Xu, Karan Shah, Antoine Ratouchniak, Jinhan Zhang, Chris Xujin Liu, Leyao Yu, Beatrice Fumagalli, and Erika Jensen. I am also grateful to other lab members: Ziming Qiu, Zhipeng Fan, Yixiang Mao, Zhiqi Chen, Jacky Yuan, Nikola Janjušević, Yueyu Hu, Haoyang Pei, Ran Gong, and Tingyu Fan.

Last but certainly not least, I would like to thank my family and my fiancée, Shuchen, for their unwavering love, support, and encouragement throughout my years of study. Their presence gave me strength, and this accomplishment would not have been possible without them.

Xupeng Chen
May 2025

To my family, for your unconditional love and support.

To my fiancée, Shuchen,
for walking with me through every step of this journey.
You are my sunshine.

ABSTRACT

Neural Speech Decoding and Understanding Leveraging Deep Learning and Speech Synthesis

by

Xupeng Chen

Advisor: Prof. Yao Wang, Prof. Adeen Flinker

Submitted in Partial Fulfillment of the Requirements for

the Degree of Doctor of Philosophy (Electrical and Computer Engineering)

May 2025

Decoding human speech from neural signals is essential for brain-computer interface (BCI) technologies, restoring communication in individuals with neurological deficits. However, this remains a highly challenging task due to the scarcity of paired neural-speech data, signal complexity, high dimensionality, and the limited availability of public tools. We first present a deep learning-based framework comprising an ECoG Decoder that translates electrocorticographic (ECoG) signals from the cortex into interpretable speech parameters, and a novel source-filter-based speech synthesizer that reconstructs spectrograms from those parameters. A companion audio-to-audio auto-encoder provides reference features to support decoder training. This framework generates naturalistic and reproducible speech and generalizes across a cohort of 48 participants. Among the tested architectures, the 3D ResNet achieved the best decoding performance in terms of Pearson Correlation Coefficient (PCC=0.804), followed closely by a SWIN model (PCC=0.796). Our models decode speech

with high correlation even under causal constraints, supporting real-time applications. We successfully decoded speech from participants with either left and right hemisphere coverage, which may benefit patients with unilateral cortical damage. We further perform occlusion analysis to identify cortical regions relevant to decoding.

We next investigate decoding from different forms of intracranial recordings, including surface (ECoG) and depth (stereotactic EEG or sEEG) electrodes, to generalize neural speech decoding across participants and diverse electrode modalities. Most prior works are constrained to 2D grid-based ECoG data from a single patient. We aim to design a deep-learning model architecture that can accommodate variable electrode configurations, support training across multiple subjects without subject-specific layers, and generalize to unseen participants. To this end, we propose SwinTW, a transformer-based model that leverages the 3D spatial locations of electrodes rather than relying on a fixed 2D layout. Subject-specific models trained on low-density 8×8 ECoG arrays outperform prior CNN and transformer baselines (PCC=0.817, N=43). Incorporating additional electrodes—including strip, grid, and depth contacts—further improves performance (PCC=0.838, N=39), while models trained solely on sEEG data still achieve high correlation (PCC=0.798, N=9). A single multi-subject model trained on data from 15 participants performs comparably to individual models (PCC=0.837 vs. 0.831) and generalizes to held-out participants (PCC=0.765 in leave-one-out validation). These results demonstrate SwinTW's scalability and flexibility, particularly for clinical settings where only depth electrodes—commonly used in chronic neurosurgical monitoring—are available. The model's ability to learn from and generalize across diverse neural data sources suggests that future speech prostheses may be trained on shared acoustic-neural corpora and applied to patients lacking direct training data.

We further investigate two complementary latent spaces for guiding neural speech decoding to enhance interpretability and structure in decoding further. HuBERT offers a discrete, phoneme-aligned latent space learned via self-supervised objectives. Decoding sEEG signals into the HuBERT token space improves intelligibility by leveraging pretrained linguistic priors. In contrast, the articulatory space provides a continuous, interpretable embedding grounded in vocal tract dynamics. The articulatory space enables speaker-specific speech synthesis through differentiable articulatory vocoders and is especially suited for both sEEG and sEMG decoding, where signals reflect muscle movements linked to articulation. While HuBERT emphasizes linguistic structure, the articulatory space provides physiological interpretability and individual control, making them complementary in design and application. We demonstrate that both spaces can serve as intermediate targets for speech decoding across invasive and non-invasive modalities. As a future direction, we extend our articulatory-guided framework toward sentence-level sEMG decoding and investigate phoneme classifiers within articulatory space to assist decoder training. These developments and the design of more advanced single- and cross-subject models support our long-term goal of building accurate, interpretable, and clinically deployable speech neuroprostheses.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ECoG	Electrocorticography
sEEG	Stereoelectroencephalography
sEMG	Surface Electromyography
EMA	Electromagnetic Articulography
PCA	Principle Component Analysis
t-SNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
HuBERT	Hidden-Unit BERT
PCC	Pearson Correlation Coefficient
STOI	Short-Time Objective Intelligibility
LSTM	Long Short-Term Memory
RNN	Recurrent neural network
MFCC	Mel Frequency Cepstral Coefficients
MNI	Montreal Neurological Institute
WER	Word Error Rate
PER	Phoneme Error Rate

Chapter 1

Introduction

1.1 Background and Related Work

Speech loss due to neurological deficits is a severe disability that limits social and work life. Advances in machine learning and Brain-computer interface (BCI) systems have pushed the envelope to develop neural speech prostheses to enable people with speech loss to communicate [1–5]. An effective modality for acquiring data to develop such decoders involves Electrocorticographic (ECoG) recordings obtained in epilepsy surgery patients [4–10]. Implanted electrodes in Epilepsy patients provides a rare opportunity to collect cortical data during speech with high spatial and temporal resolution. Such approaches have revealed promising results in speech decoding [4, 5, 8–11].

Two challenges are inherent to speech decoding from neural signals. First, the data to train neural-to-speech decoding is limited in duration, while deep learning models require extensive training data. Second, speech production varies in rate, intonation, pitch, etc., even within a single speaker producing the same word, complicating the underlying model representation [12, 13]. These

challenges have led to diverse approaches to speech decoding with limited consistency in model approaches and the availability of public code to test and replicate findings across research groups.

This dissertation is driven by a central research question: ***how to decode neural signals into realistic and intelligible speech***. To address this challenge, we systematically investigate how to establish an effective common latent representation between brain and speech signals, which enables overcoming major obstacles such as data scarcity and speech variability. In this section, we summarize the main contributions of this work within the context of prior research and current advances in the field.

1.2 Main Contributions

Beyond the speech synthesizer developed by our team (Chapter 3), significant strides have recently been made in developing high-performance and intelligible speech prostheses [14]. These systems have demonstrated the potential to restore communication abilities in individuals with paralysis, leveraging high-density electrocorticography (ECoG) and Utah array neural recordings, and marking remarkable progress in neural speech decoding.

While these advances benefit from the high spatial resolution of invasive recordings, their clinical and research utility is constrained by the invasiveness of the procedures. In contrast, stereoelectroencephalography (sEEG) offers a less invasive alternative that is increasingly accessible in clinical settings. Recent work [15] has shown that imagined speech can be synthesized in real time from sEEG signals, but the use of simplistic decoding pipelines that disregard temporal dependencies has so far yielded low-quality, unintelligible speech.

Flow-based models such as WaveGlow, recently employed for offline synthesis in sEEG-based speech decoding [16], hold promise as vocoders for real-time brain-computer interface (BCI) systems due to their fast inference speeds. However, intelligible speech decoding from sEEG remains an open challenge, and it remains unclear whether this is due to insufficient information in sEEG recordings or the limitations of current decoding frameworks.

Importantly, despite the progress in ECoG- and sEEG-based speech prostheses, most existing efforts have relied on *subject-specific models*, with no prior work developing cross-subject decoding frameworks using either ECoG or sEEG. Moreover, while recent studies such as [14] have incorporated HuBERT-based representations [17], the potential of such self-supervised representations, as well as physiologically interpretable alternatives such as articulatory space [18], remains underexplored. Finally, an important question is whether non-invasive, articulation-related signals, such as surface electromyography (sEMG), can be leveraged to achieve effective speech decoding.

This dissertation is driven by the central research question: ***how to decode neural signals into realistic and intelligible speech***. To address this challenge, we systematically investigate latent representations that effectively bridge neural and speech signals and explore their applicability across diverse modalities and populations. The main contributions of this work are summarized as follows:

- In our previous work [11], we employed a VAE-GAN architecture to map ECoG signals into a latent representation regularized by a GAN model. While the model successfully reconstructed realistic spectrograms in perception tasks, it struggled in production tasks, often yielding blurred outputs.

- Inspired by [19], we developed a differentiable source-filter speech synthesizer, described in Chapter 3, enabling the re-synthesis of highly intelligible and realistic speech even with limited data. We substantially improved decoding performance by mapping ECoG signals to compact and physically interpretable speech parameters [20]. We extended the framework to speech production tasks and demonstrated robust decoding in participants with low-density electrode arrays, irrespective of hemispheric implantation.

- In Chapter 4, we advanced cross-subject decoding by integrating surface (ECoG) and depth (sEEG) modalities through the development of SwinTW [21]. This multi-modal, cross-subject framework enhanced the robustness and generalizability of speech decoding across participants.

- In Chapter 5, we investigate the effectiveness of two alternative latent representations for speech decoding, focusing on quantized latent spaces learned through masked encoding and clustering (HuBERT [17]), alongside physiologically interpretable, speaker-specific representations (Articulatory Space [18]). Both of these latent representations approach improved neural speech decoding performance.

- In Chapter 6, we extended the articulatory space-based framework to non-invasive modalities, specifically sEMG-based speech decoding, and demonstrated highly promising performance. We additionally train an auxiliary phoneme classifier, which may serve as a foundation for future efforts to enhance speech decoding performance.

Chapter 2

Background and Related Work

2.1 Related Work

Speech loss due to neurological deficits is a severe disability that limits social and work life. Advances in machine learning and Brain-computer interface (BCI) systems have pushed the envelope to develop neural speech prostheses to enable people with speech loss to communicate [1–5]. An effective modality for acquiring data to develop such decoders involves Electrocorticographic (ECoG) recordings obtained in epilepsy surgery patients [4–10]. Implanted electrodes in Epilepsy patients provides a rare opportunity to collect cortical data during speech with high spatial and temporal resolution. Such approaches have revealed promising results in speech decoding [4,5,8–11].

Two challenges are inherent to speech decoding from neural signals. First, data to train personalized neural-to-speech decoding models are limited in duration, while deep learning models require extensive training data. Second, speech production varies in rate, intonation, pitch, etc., even within a single

speaker producing the same word, complicating the underlying model representation [12, 13]. These challenges have led to diverse speech decoding approaches with various model architectures. Currently, there is a limited availability of public code to test and replicate findings across research groups.

Earlier approaches to decode and synthesize speech spectrograms from neural signals focused on linear models. These approaches achieved $\sim$ 0.6 Pearson correlation coefficient (PCC) or lower while providing simple model architectures that are easy to interpret and do not require large training data-sets [22–24]. Recent research has focused on deep neural networks leveraging convolutional [8,9] and recurrent [5,10,25] network architectures. These approaches vary across two major dimensions: the intermediate latent representation used to model speech and the speech quality produced after synthesis. For example, cortical activity has been decoded into an articulatory movement space, which is then transformed into speech, providing robust decoding performance but with a non-natural synthetic voice reconstruction [25]. Conversely, some approaches produced naturalistic reconstruction leveraging wavenet vocoders [8], General Adversarial Networks [11], and unit selection [26], but with limited accuracy. A recent study in one implanted patient [14], provided both robust accuracies combined with a naturalistic speech waveform by leveraging the quantized HuBERT features [17] as an intermediate representation space and a pretrained speech synthesizer which converts the HuBERT features into speech. However, HuBERT features do not carry speaker-dependent acoustic information and thus can only be used to generate a generic speaker's voice, requiring a separate model to translate the generic voice to a specific patient's voice. Further, this study and most previous approaches employ non-causal architectures, which could limit real-time applications that typically require causal operations.

2.2 Evaluation Metrics

2.2.1 Speech based evaluation metrics

Following [8, 20, 25, 27], we used three metrics to evaluate the speech decoding performance for aligned original and decoded speech. The metrics are used in Chapter. 3, 4, 5, and 6

1) Pearson Correlation Coefficient (PCC) measures the normalized correlation between the decoded spectrogram and the actual spectrogram. It is a widely used metric to evaluate the accuracy of the decoded spectrogram.

2) STOI+ [28] is another metric that measures the similarity between decoded and original speech. STOI+ has been reported to have a monotonic relationship with speech intelligibility. The STOI+ value ranges from -1 to 1, and higher STOI+ indicates better intelligibility.

3) Mel-cepstral distortion (MCD) [29] measures the differences between 25 acoustic features generated from the decoded speech and the original speech. A lower MCD is better. MCD is calculated as follows:

$$\mathrm{MCD} = \frac{10}{\ln(10)} \sqrt{\sum_{0<d<25} (\mathrm{mc}_d - \hat{\mathrm{mc}}_d)^2}$$

where mc_d and $\hat{\mathrm{mc}}_d$ denote the d-th feature generated from the original and decoded speech.

2.2.2 Automatic Speech Recognition based Evaluation Metrics

Here, we adopt a state-of-the-art automatic speech recognition (ASR) model HuBERTCTC [17] (`facebook/hubert-large-ls960-ft` from huggingface [30]), to

transcribe from the audio of each trial. Then, we could calculate the following metrics: edit distance and decoding accuracy.

Edit Distance Here, we use the Levenshtein distance [31] to calculate the distance between the transcribed words of the original and decoded speeches. It is a string metric used to measure the difference between two sequences. The Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into another. The fewer edits it takes, the closer the two sentences are semantically. When the decoded speech is perfectly recognized, the edit distance is zero. Note that sometimes perceptually recognizable speech may not be transcribed accurately by the ASR model. The edit distance only reflects a lower performance limit of the sEEG decoding model.

To further quantify decoding performance at different linguistic levels, we could compute the Word Error Rate (**WER**) and Phone Error Rate (**PER**) based on the same Levenshtein alignment. Let S, D, and I denote the numbers of substitutions, deletions, and insertions, and let N_{words} and N_{phones} be the total numbers of words and phones in the reference transcription. We define:

$$\text{WER} = \frac{S + D + I}{N_{\text{words}}}, \tag{2.1}$$

$$\text{PER} = \frac{S + D + I}{N_{\text{phones}}}. \tag{2.2}$$

Closed-set Word Matching Accuracy In early tests we tried the standard open vocabulary word-error rate (WER), but it quickly proved unsuitable for our setting. WER counts substitutions, deletions, and insertions *between words*, so it is meaningful only when the output is a multi-word sentence. Our task, by

contrast, decodes a single isolated word; with just one token the notions of "extra," "missing," or "substituted" words collapse, making WER undefined (or trivially $0/100\%$). We therefore abandon WER and adopt metrics tailored to single-word decoding. To address these limitations and obtain a more reliable and interpretable measure, we constrained evaluation to a closed vocabulary V of size $|V| = 50$, which exactly matches the fifty words in our test set, and report closed-set word matching accuracy.

Let there be N trials, and for trial i let the true target word be $w_i \in V$. The ASR model produces a raw transcription string $\hat{w}_i$ from decoded audio. We compute the Levenshtein distance

$$d(\hat{w}_i, w) \quad \forall\, w \in V,$$

and select the nearest neighbor

$$w_i^* = \arg\min_{w \in V} d(\hat{w}_i, w).$$

We count trial i as correct if $w_i^* = w_i$. The closed-set matching accuracy is then

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{w_i^* = w_i\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. In this 50-class forced-choice setting, the chance level is $1/50 = 2\%$, and the metric avoids the instability of open-vocabulary WER on single-word trials.

Chapter 3

A Neural Speech Decoding Framework Leveraging Deep Learning and Source-Filter Based Speech Synthesis

3.1 Introduction

Building on the challenges outlined in Chapter. 2, namely limited per-subject data, speaker-dependent acoustic variation, and the need for causal inference, our framework consists of an ECoG Decoder that maps the ECoG signals to interpretable acoustic speech parameters and a Source-Filter based Speech Synthesizer that translates the speech parameters to a spectrogram. The Speech Synthesizer is differentiable, enabling us to minimize the spectrogram reconstruction error during the training of the ECoG Decoder. The low-dimensional latent space describing the speech parameters, together with the use of the speech parameters generated by a pre-trained Speech Encoder as a guidance for the ECoG

decoder training, overcomes data scarcity issues. Our publicly available framework produces naturalistic speech highly resembling the speaker's own voice, and the ECoG decoder can be realized with different deep-learning model architectures and using different causality directions. Here, we report this framework with multiple deep architectures (i.e., convolutional, recurrent, transformer) as the ECoG decoder and apply it to 48 neurosurgical patients. Our framework performs with high accuracy across models, with the best performance obtained by the convolutional (ResNet) architecture (Pearson Correlation of 0.806 between the original and decoded spectrogram). Our framework can achieve high accuracy using only causal processing and relatively low spatial sampling on the cortex. Further, we show comparable speech decoding from the grid implants on the left and right hemispheres, providing a proof of concept for neural prosthetics in patients suffering from expressive Aphasia (damage limited to the left hemisphere), albeit such an approach must be tested in patients with damage to the left hemisphere. Finally, we provide a publicly available neural decoding pipeline (`https://github.com/flinkerlab/neural_speech_decoding`) that offers flexibility in ECoG decoding architectures, to push forward research across the speech science and prostheses communities.

3.2 Proposed Framework

Our ECoG-to-Speech framework consists of an ECoG Decoder and a Speech Synthesizer, shown in the upper part of Fig. 3.1. The neural signals are fed into an ECoG Decoder that generates speech parameters, followed by a Speech Synthesizer, which translates the parameters into spectrograms (which are then converted to waveform by the Griffin-Lim algorithm [32]). The training of our framework takes two steps. We first use semi-supervised learning on the speech

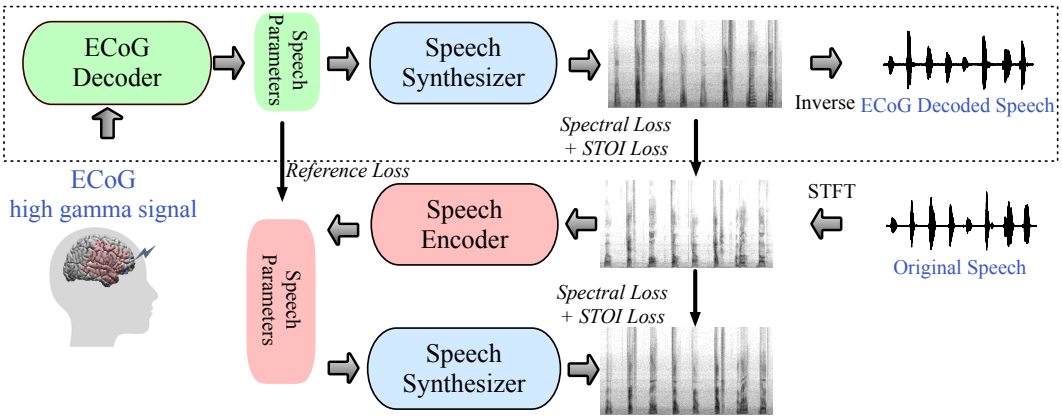


FIGURE 3.1: | **The proposed neural speech decoding framework.** The upper part shows the ECoG-to-speech decoding pipeline. The ECoG Decoder generates time-varying speech parameters from ECoG signals. The Speech Synthesizer generates spectrograms from the speech parameters. A separate spectrogram inversion algorithm converts the spectrograms to speech waveforms. The lower part shows the speech-to-speech auto-encoder that generates the guidance for the speech parameters to be produced by the ECoG Decoder during the ECoG Decoder training. The Speech Encoder maps an input spectrogram to speech parameters, which are then fed to the same Speech Synthesizer to reproduce the spectrogram. The Speech Encoder and a few learnable participant-specific parameters in the Speech Synthesizer are pre-trained using speech signals only. Only the upper part is needed for decoding the speech from ECoG signals once the pipeline is trained.

signals alone. An auto-encoder, shown in the lower part of Fig. 3.1, is trained so that the Speech Encoder derives speech parameters from a given spectrogram, while the Speech Synthesizer (used here as the decoder) reproduces the spectrogram from the speech parameters. Our Speech Synthesizer is fully differentiable and generates speech through a weighted combination of voice and unvoiced speech components generated from input time series of speech parameters, including pitch, formant frequencies, loudness, etc. The Speech Synthesizer has only a few subject-specific parameters that are learned as part of the auto-encoder training. (see Methods section: Speech Synthesizer for more details). Currently, our Speech Encoder and Speech Synthesizer are subject-specific and can be trained using any speech signal of a participant, not limited to those with corresponding ECoG signals.

In the next stage, we train the ECoG Decoder in a supervised manner based on ground-truth spectrograms (using measures of spectrogram difference and Short-Time Objective Intelligibility (STOI) [8, 33]) as well as guidance for the speech parameters generated by the pre-trained Speech Encoder (i.e., reference loss between speech parameters). By limiting the number of speech parameters (18 at each time step, see Methods Section: Summary of Speech Parameters) and using the reference loss, the ECoG Decoder can be trained with limited corresponding ECoG and speech data. Further, because our Speech Synthesizer is differentiable, we can back-propagate the spectral loss (differences between the original and decoded spectrograms) to update the ECoG Decoder. Here, we provide multiple ECoG Decoder architectures to choose from, including 3D ResNet [34], 3D SWIN Transformer [35], and LSTM [36]. Importantly, unlike many methods in the literature, we employ ECoG Decoders that can operate in a causal manner, which is necessary for real-time speech generation from neural signals. Note that once the ECoG Decoder and Speech Synthesizer are trained,

they can be used for ECoG to speech decoding without using the Speech Encoder.

3.3 Methods

3.3.1 Source-Filter Based Speech Synthesizer

Our Speech Synthesizer is inspired by the traditional speech vocoder, which generates speech by switching between voice and unvoiced content, each generated by filtering a specific excitation signal. Instead of switching between the two components, we use a soft mix of the two components, making the Speech Synthesizer differentiable. This enables us to train the ECoG Decoder and the Speech Encoder end-to-end by minimizing the spectrogram reconstruction loss with back-propagation. Our Speech Synthesizer can generate a spectrogram from a compact set of speech parameters, enabling the training of the ECoG Decoder with limited data. As shown in Fig. 3.2, the synthesizer takes dynamic speech parameters as input and contains two pathways. The voice pathway applies a set of formant filters (each specified by center frequency f_i^t, bandwidth b_i^t that is dependent on f_i^t, and amplitude a_i^t) to the harmonic excitation (with pitch frequency f_0) and generates the voice component, $V^t(f)$, for each time step t and frequency f. The noise pathway filters the input white noise with an unvoiced filter (consisting of a broadband filter defined by center frequency $f_{\hat{u}}^t$, bandwidth $b_{\hat{u}}^t$, and amplitude $a_{\hat{u}}^t$ and the same six formant filters used for the voice filter) and produces the unvoiced content, $U^t(f)$. The synthesizer combines the two components with a voice weight $\alpha^t \in [0, 1]$ to obtain the combined spectrogram $\widetilde{S}^t(f)$ as follows:

$$\widetilde{S}^t(f) = \alpha^t V^t(f) + (1 - \alpha^t) U^t(f).$$

The factor α^t acts as a soft switch for the gradient to flow back through the synthesizer. The final speech spectrogram is

$$\widehat{S}^t(f) = L^t \widetilde{S}^t(f) + B(f),$$

where L^t is the loudness modulation and $B(f)$ is the background noise. We describe the various components in more detail below.

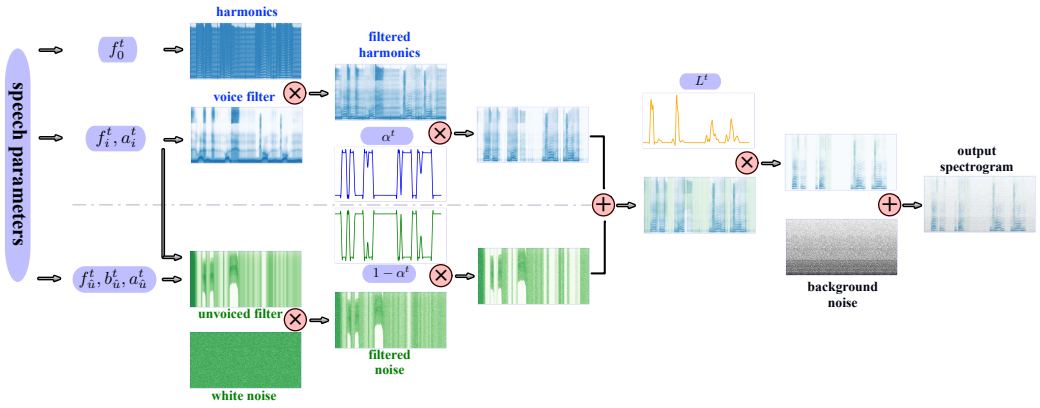


FIGURE 3.2: | **Differentiable Speech Synthesizer architecture.** Our Speech synthesizer generates the spectrogram at time t by combining a voice component and an unvoiced component based on a set of speech parameters at t. The upper part represents the voice pathway, which generates the voice component by passing a harmonic excitation with fundamental frequency f_0^t through a voice filter (which is the sum of 6 formant filters, each specified by formant frequency f_i^t and amplitude a_i^t). The lower part describes the noise pathway, which synthesizes the unvoiced sound by passing white noise through an unvoiced filter (consisting of a broadband filter defined by center frequency $f_{\hat{u}}^t$, bandwidth $b_{\hat{u}}^t$, and amplitude $a_{\hat{u}}^t$ and the same six formant filters used for the voice filter). The two components are next mixed with voice weight α^t and unvoiced weight $1-\alpha^t$, respectively, and then amplified by loudness L^t. A background noise (defined by a stationary spectrogram $B(f)$) is finally added to generate the output spectrogram. There are a total of 18 speech parameters at any time t, indicated in purple boxes.

Formant filters in the voice pathway

We use multiple formant filters in the voice pathway to model formants that represent vowels and nasal information. The formant filters capture the resonance in the vocal tract, which can help recover a speaker's timbre characteristics and generate natural-sounding speech. We assume the filter for each formant is time-varying and can be derived from a prototype filter $G_i(f)$, which achieves maximum at a center frequency f_i^{proto} and has a half-power bandwidth b_i^{proto}. The prototype filters have learnable parameters and will be discussed later. The actual formant filter at any time is written as a shifted and scaled version of $G_i(f)$. Specifically, at time t, given an amplitude $\left(a_i^t\right)$, a center frequency $\left(f_i^t\right)$, and a bandwidth $\left(b_i^t\right)$, the frequency domain representation of the i-th formant filter is

$$F_i^t(f) = a_i^t \cdot G_i\left(\frac{b_i^{\text{proto}}}{b_i^t} \cdot \left(f - f_i^t\right) + f_i^{\text{proto}}\right), f \in [0, f_{\max}] \tag{3.1}$$

where $f_{\max}$ is half of the speech sampling frequency, which in our case is 8000 Hz.

Rather than letting the bandwidth parameters b_i^t be independent variables, based on the empirically observed relationships between b_i^t and the center frequencies f_i^t, we set

$$b_i^t = \begin{cases} a\left(f_i^t - f_\theta\right) + b_0, & \text{if } f_i^t > f_\theta \\ b_0, & \text{otherwise} \end{cases} \tag{3.2}$$

The threshold frequency f_θ, slope a, and baseline bandwidth b_0 are three parameters that are learned during the auto-encoder training, shared among all six formant filters. This parameterization helps to reduce the number of speech parameters to be estimated at every time sample, making the representation space more compact.

Finally the filter for the voice pathway with N formant filters is given by $F_{\mathrm{v}}^t(f) = \sum_{i=1}^{N} F_i^t(f)$. Previous studies have shown that two formants (N=2) are enough for intelligible reconstruction [37], but we use N=6 for more accurate synthesis in our experiments.

unvoiced filters

We construct the unvoiced filter by adding a single broadband filter $F_{\hat{u}}^t(f)$ to the formant filters for each time step t. The broadband filter $F_{\hat{u}}^t(f)$ has the same form as equation (3.1) but has its own learned prototype filter $G_{\hat{u}}(f)$ The speech parameters corresponding to the broadband filter include $\left(\alpha_{\hat{u}}^t, f_{\hat{u}}^t, b_{\hat{u}}^t\right)$. We do not impose a relationship between the center frequency $f_{\hat{u}}^t$ and the bandwidth $b_{\hat{u}}^t$. This allows more flexibility in shaping the broadband unvoiced filter. But we constrain $b_{\hat{u}}^t$ to be larger than 2000 Hz to capture the wide spectral range of obstruent phonemes. Instead of using only the broadband filter, we also retain the N formant filters in the voice pathway F_i^t for the noise pathway. This is based on the observation that humans perceive consonants such as /p/ and /d/ not only by their initial bursts but also by their subsequent formant transitions until the next vowel [38]. We use identical formant filter parameters to encode these transitions. The overall unvoiced filter is: $F_{\mathrm{u}}^t(f) = F_{\hat{u}}^t(f) + \sum_{i=1}^{N} F_i^t(f)$.

Voice Excitation

We use the voice filter in the voice pathway to modulate the harmonic excitation. Following [19], we define the harmonic excitation as $h^t = \sum_{k=1}^{K} h_k^t$, where $K = 80$ is the number of harmonics.

The value of the k-th resonance at time step t is $h_k^t = \sin(2\pi k \phi^t)$ with $\phi^t = \sum_{\tau=0}^{t} f_0^\tau$, where f_0^τ is the fundamental frequency at time τ. The spectrogram of

h^t forms the harmonic excitation in the frequency domain $H^t(f)$, and the voice excitation is $V^t(f) = F_{\mathrm{v}}^t(f)H^t(f)$.

Noise Excitation

The noise pathway models consonant sounds (plosives and fricatives). It is generated by passing a stationary Gaussian white noise excitation through the unvoiced filter. We first generate the noise signal $n(t)$ in the time domain by sampling from the Gaussian process $\mathcal{N}(0,1)$ and then obtain its spectrogram $N^t(f)$. The spectrogram of the unvoiced component is $U^t(f) = F_u^t(f)N^t(f)$.

Summary of speech parameters

The synthesizer generates the voice component at time t by driving a harmonic excitation with pitch frequency f_0^t through N formant filters in the voice pathway, each described by two parameters (f_i^t, a_i^t). The unvoiced component is generated by filtering a white noise through the unvoiced filter consisting of an additional broadband filter with three parameters $(f_{\hat{u}}^t, b_{\hat{u}}^t, a_{\hat{u}}^t)$. The two components are mixed based on the voice weight α^t and further amplified by the loudness value L^t. In total, the synthesizer input includes 18 speech parameters at each time step.

Unlike DDSP in [19], we do not directly assign amplitudes to the K harmonics. Instead, the amplitude in our model depends on the formant filters, which has two benefits:

- **The representation space is more compact.** DDSP requires 80 amplitude parameters a_k^t for each of the 80 harmonic components f_k^t $(k = 1, 2, \ldots, 80)$ at each time step. In contrast, our synthesizer only needs a total of 18 parameters.

- **The representation is more disentangled.** For human speech, the vocal tract shape (affecting the formant filters) is largely independent of the vocal cord tension (which determines the pitch). Modeling these two separately leads to a disentangled representation.

  In contrast, DDSP specifies the amplitude for each harmonic component directly resulting in entanglement and redundancy between these amplitudes. Furthermore, it remains uncertain whether the amplitudes a_k^t could be effectively controlled and encoded by the brain. In our approach, we explicitly model formant filters and fundamental frequency, which possess clear physical interpretations and are likely to be directly controlled by the brain. Our representation also enables a more robust and direct estimation of the pitch.

Speaker-Specific Synthesizer Parameters

Prototype filters: Instead of using a pre-determined prototype formant filter shape, e.g., a standard Gaussian function, we learn a speaker-dependent prototype filter for each formant to allow more expressive and flexible formant filter shapes. We define the prototype filter $G_i(f)$ of the i-th formant as a piece-wise linear function, linearly interpolated from $g_i[m], m = 1 \dots M$, the amplitudes of the filter at M uniformly sampled frequencies in the range $[0, f_{\max}]$. We constrain $\mathrm{g}_i[\mathrm{m}]$ to increase and then decrease monotonically so that $G_i(f)$ is uni-modal and has a single peak value of 1. Given $g_i[m], m = 1 \dots M$, we can determine the peak frequency f_i^{proto} and the half-power bandwidth b_i^{proto} of $G_i(f)$.

The prototype parameters $g_i[m], m = 1 \dots M$ of each formant filter are time-invariant and are determined during the auto-encoder training. Compared with

[39], we increase M from 20 to 80 to enable more expressive formant filters, essential for synthesizing male speakers' voices.

We similarly learn a prototype filter for the broadband filter $G_{\hat{u}}(f)$ for the unvoiced component, which is specified by M parameters $g_{\hat{u}}(m)$.

Background Noise: The recorded sound typically contains background noise. We assume the background noise is stationary and has a specific frequency distribution, depending on the speech recording environment. This frequency distribution $B(f)$ is described by K parameters, where K is the number of frequency bins (K=256 for females and 512 for males). The K parameters are also learned during the auto-encoder training. The background noise is added to the mixed speech components to generate the final speech spectrogram.

To summarize, our Speech Synthesizer has the following learnable parameters: the $M = 80$ prototype filter parameters for each of the $N = 6$ formant filters and the broadband filters (totaling $M(N+1) = 560$), the three parameters f_θ, a, b_0 relating the center frequency and bandwidth for the formant filters (totaling 21), and K parameters for the background noise (256 for female and 512 for male). The total number of parameters for female speakers is 837, and that for male speakers is 1093. Note that these parameters are speaker-dependent but time-independent and can be learned together with the Speech Encoder during the training of the speech-to-speech auto-encoder, using the speaker's speech only.

3.3.2 Speech Encoder

The Speech Encoder extracts a set of (18) speech parameters at each time point from a given spectrogram, which are then fed to the speech synthesizer to reproduce the spectrogram.

We use a simple network architecture for the Speech Encoder, with temporal convolutional layers and multilayer perceptron (MLP) across channels at the same time point, as shown in Fig. 3.3**a**. We encode pitch f_0^t by combining features generated from linear and mel-scale spectrograms. The other 17 speech parameters are derived by applying temporal convolutional layers and channel MLP to the linear scale spectrogram. To generate formant filter center frequencies $f^t_{i=1 \text{ to } 6}$, broadband unvoiced filter frequency $f^t_{\hat{u}}$, and pitch f_0^t, we use sigmoid activation at the end of the corresponding channel MLP to map the output to $[0, 1]$, and then de-normalize it to real values by scaling $[0, 1]$ to predefined $[f_{min}, f_{max}]$. The $[f_{min}, f_{max}]$ values for each frequency parameter are chosen based on previous studies [40–43]. Our compact speech parameter space facilitates stable and easy training of our Speech Encoder. Models were coded using Pytorch version 1.21.1 in Python.

3.3.3 ECoG Decoder

We present the design details of three ECoG Decoders: the 3D ResNet ECoG Decoder, the 3D SWIN Transformer ECoG Decoder, and the LSTM ECoG Decoder. Models were coded using Pytorch version 1.21.1 in Python.

3D ResNet ECoG Decoder

This decoder adopts the ResNet architecture [34] for the feature extraction backbone of the decoder. Fig. 3.3**b** illustrates the feature extraction part. The model views the ECoG input as 3D tensors with spatiotemporal dimensions. In the first layer, we apply only temporal convolution to the signal from each electrode because the ECoG signal exhibits more temporal than spatial correlations. In the

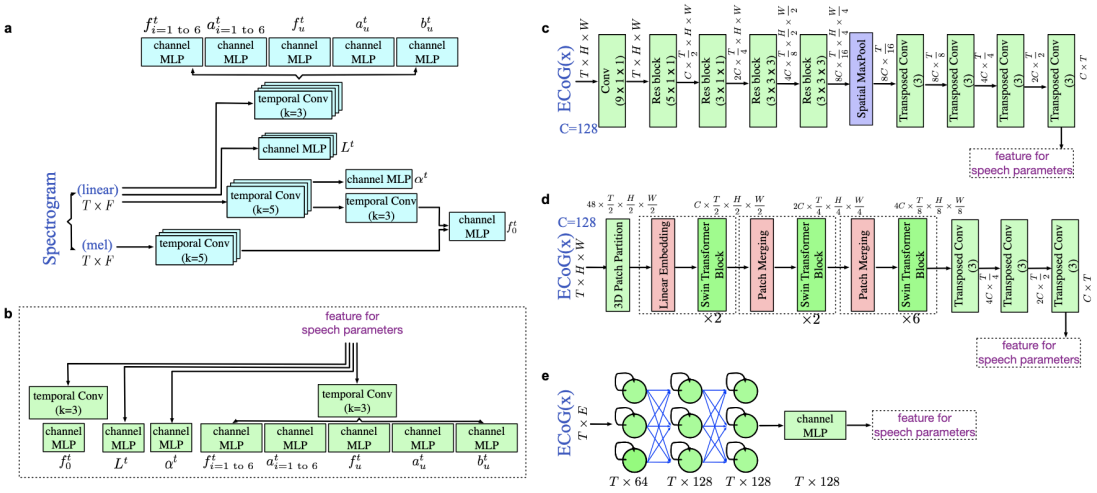


FIGURE 3.3: | **Speech Encoder and ECoG Decoder.** **a,** Speech Encoder architecture. We input a spectrogram into a network of temporal convolution layers and channel MLPs that produce speech parameters. **c,** ECoG Decoder using the 3D ResNet architecture. We first use several temporal and spatial convolutional layers with residual connections and spatiotemporal pooling to generate down-sampled latent features and then use corresponding transposed temporal convolutional layers to up-sample the features to the original temporal dimension. We then apply temporal convolution layers and channel MLPs to map the features to speech parameters, as shown in (**b**). The non-causal version uses non-causal temporal convolution in each layer, whereas the causal version uses causal convolution. **d,** ECoG Decoder using the 3D SWIN architecture. We use three or four stages of 3D SWIN blocks with spatial-temporal attention (3 blocks for LD and 4 blocks for HB) to extract the features from the ECoG signal. We then use the transposed versions of temporal convolution layers as in (**c**) to up-sample the features. The resulting features are mapped to the speech parameters using the same structure shown in (**b**). Non-causal versions apply temporal attention to past, present, and future tokens, whereas the causal version only applies temporal attention to past and present tokens. **e,** ECoG Decoder using LSTM layers. We use three LSTM layers and one layer of channel MLP to generate features. We then reuse the prediction layers in (**b**) to generate corresponding speech parameters. The non-causal version employs bi-directional LSTM in each layer, whereas the causal version uses uni-directional LSTM.

subsequent parts of the decoder, we have four residual blocks that extract spatiotemporal features using 3D convolution. After down-sampling the electrode dimension to 1×1 and the temporal dimension to $T/16$, we use several transposed Conv layers to up-sample the features to the original temporal size T. Fig. 3.3**c** shows how to generate the different speech parameters from the resulting features using different temporal convolution and channel MLP layers. The temporal convolution operation can be causal (i.e., using only past and current samples as input) or non-causal (i.e., using past, current, and future samples), leading to causal and non-causal models.

3D Swin Transformer ECoG Decoder

Swin Transformer [35] employs the window and shift window methods to enable self-attention of small patches within each window. This reduces the computational complexity and introduces the inductive bias of locality. Since our ECoG input data has three dimensions, we extend the Swin Transformer to three dimensions to enable local self-attention in both temporal and spatial dimensions among 3D patches. The local attention within each window gradually becomes global attention as the model merges neighboring patches in deeper Transformer stages.

Fig. 3.3**d** illustrates the overall architecture of the proposed 3D Swin Transformer. The input ECoG signal has a size of $T \times H \times W$, where T is the number of frames and $H \times W$ is the number of electrodes at each frame. We treat each 3D patch of size $2 \times 2 \times 2$ as a token in the 3D Swin Transformer. The 3D patch partitioning layer produces $\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2}$ 3D tokens, each with a $C = 48$ dimensional feature. A linear embedding layer then projects the features of each token to a higher dimension $C(= 128)$.

The 3D Swin Transformer comprises three stages with $2, 2, 6$ layers, respectively, for LD participants and four stages with $2, 2, 6, 2$ layers for HB participants. It performs $2 \times 2 \times 2$ spatial and temporal down-sampling in the patch merging layer of each stage. The patch merging layer concatenates the features of each group of $2 \times 2 \times 2$ temporally and spatially adjacent tokens. It applies a linear layer to project the concatenated features to $\frac{1}{4}$ of their original dimension after merging. In the 3D Swin Transformer block, we replace the multi-head self-attention (MSA) module in the original Swin Transformer with the 3D shifted window multi-head self-attention module. It adapts the other components to 3D operations as well. A Swin Transformer block consists of a 3D-shifted window-based MSA module followed by a feed-forward network (FFN), a 2-layer MLP. Layer Normalization (LN) is applied before each MSA module and FFN, and a residual connection is applied after each module.

Consider a stage with $T \times H \times W$ input tokens. If the 3D window size is $P \times M \times M$, we partition the input into $\lceil \frac{T}{P} \rceil \times \lceil \frac{H}{M} \rceil \times \lceil \frac{W}{M} \rceil$ non-overlapping 3D windows evenly. We choose $P = 16, M = 2$. We perform the multi-head self-attention within each 3D window. However, this design lacks connection across adjacent windows, which may limit the representation power of the architecture. Therefore, we extend the shifted 2D window mechanism of the Swin Transformer to shifted 3D windows. In the second layer of the stage, we shift the window partition configuration by $(\frac{P}{2}, \frac{M}{2}, \frac{M}{2})$ tokens along the temporal, height, and width axes from the previous layer. This creates cross-window connections for the self-attention module. This shifted 3D window design enables the interaction of electrodes with longer spatial and temporal distances by connecting neighboring tokens in non-overlapping 3D windows in the previous layer.

The temporal attention in the self-attention operation can be constrained to

be causal (i.e., each token only attends to tokens temporally before it) or non-causal (i.e., each token can attend to tokens temporally before or after it), leading to the causal and non-causal models, respectively.

LSTM Decoder

The decoder uses the LSTM architecture [44] for the feature extraction in Fig. 3.3e. Each LSTM cell is composed of a set of gates that control the flow of information: the input gate, the forget gate, and the output gate. The input gate regulates the entry of new data into the cell state, the forget gate decides what information is discarded from the cell state, and the output gate determines what information is transferred to the next hidden state and can be output from the cell.

In the LSTM architecture, the ECoG input would be processed through these cells sequentially. For each time step T, the LSTM would take the current input x_t and the previous hidden state h_{t-1} and would produce a new hidden state h_t and output y_t. This process allows the LSTM to maintain information over time and is particularly useful for tasks such as speech and neural signal processing, where temporal dependencies are critical. Here we use three layers of LSTM and one linear layer to generate features to map to speech parameters. Unlike 3D ResNet and 3D SWIN, we keep the temporal dimension unchanged across all layers.

3.3.4 Model Training

Training of the Speech Encoder and Speech Synthesizer

As described earlier, we pre-train the Speech Encoder and the learnable parameters in the Speech Synthesizer to perform a speech-to-speech auto-encoding task.

We use multiple loss terms for the training. The modified multi-scale spectral loss is inspired by [19] and defined as

$$L_{MSS}(\widehat{S}^t(f), S^t(f)) = L(\widehat{S}^t(f), S^t(f)) + L(\widehat{S}^t_{\text{mel}}(f), S^t_{\text{mel}}(f))$$

with

$$L(x, y) = \|x - y\|_1 + \|\log x - \log y\|_1.$$

Here, $S^t(f)$ denotes the ground truth spectrogram and $\widehat{S}^t(f)$ the reconstructed spectrogram in the linear scale, $S^t_{\text{mel}}(f)$ and $\widehat{S}^t_{\text{mel}}(f)$ are the corresponding spectrograms in the mel-frequency scale. We sample the frequency range $[0, 8000\text{ Hz}]$ with $K = 256$ bins for female participants. For male patients, we set $K = 512$ since they have lower f_0, and it is better to have a higher resolution in frequency.

To improve the intelligibility of reconstructed speech, we also introduce the STOI loss by implementing the STOI+ metric [28], which is a variation of the original Short-Time Objective Intelligibility (STOI) metric [8,33]. STOI+ [28] discards the normalization and clipping step in STOI and has been shown to perform best among intelligibility evaluation metrics. First, a one-third octave band analysis [33] is performed by grouping DFT bins into 15 one-third octave bands with the lowest center frequency set equal to 150 Hz and the highest center frequency equal to approximately 4.3 kHz. Let $\hat{x}(k, m)$ denote the k^{th} DFT-bin of the m^{th} frame of the ground truth speech. The norm of the j^{th} one-third octave band, referred to as a TF-unit, is then defined as:

$$X_j(m) = \sqrt{\sum_{k=k_1(j)}^{k_2(j)-1} |\hat{x}(k, m)|^2}$$

where $k_1(j)$ and $k_2(j)$ denote the one-third octave band edges rounded to the

nearest DFT-bin. The TF representation of the processed speech $\hat{y}$ is obtained similarly and denoted by $Y_j(m)$. We then extract the short-time temporal envelopes in each band and frame, denoted $X_{j,m}$ and $Y_{j,m}$, where $X_{j,m} = \left[X_j(m-N+1), X_j(m-N+2)\right.$ with $N = 30$. The STOI+ metric is the average of the Pearson correlation coefficient $d_{j,m}$ between $X_{j,m}$ and $Y_{j,m}$, overall j and m [28]:

$$STOI_{plus} = \frac{1}{JM} \sum_{j,m} d_{j,m}$$

We use the negative of the STOI+ metric as the STOI loss:

$$L_{STOI} = -STOI_{plus}$$

where J and M are the total numbers of frequency bins (J=15) and frames, respectively. Note that L_{STOI} is differentiable with respect to $\widehat{S}^t(f)$, thus can be used to update the model parameters generating the predicted spectrogram $\widehat{S}^t(f)$.

To further improve the accuracy for estimating the pitch f_0^t and formant frequencies $\widetilde{f}^t_{\text{i=1 to 4}}$, we add supervisions to them using the formant frequencies extracted by the Praat method [45]. The supervision loss is defined as

$$L_{\text{supervision}} = \|\widetilde{f}_0^t - f_0^t\|_2^2 + \sum_{i=1}^{4} \beta_i \|\widetilde{f}_i^t - f_i^t\|_2^2,$$

where the weights β_i are chosen to be $\beta_1 = 0.1, \beta_2 = 0.06, \beta_3 = 0.03, \beta_4 = 0.02$, based on empirical trials. The overall training loss is defined as:

$$L = L_{\text{MSS}} + \lambda_1 L_{\text{STOI}} + \lambda_2 L_{\text{supervision}}$$

Where weighting parameters λ_i are empirically optimized to be $\lambda_1 = 1.2, \lambda_2 =$

0.1 through testing the performances on three hybrid-density participants with different parameter choices.

Training of the ECoG Decoder

With the reference speech parameters generated by the Speech Encoder and the target speech spectrograms as ground truth, the ECoG Decoder is trained to match these targets. Let us denote the decoded speech parameters as $\widetilde{C}_j^t$, and their references as C_j^t, where j enumerates all speech parameters fed to the Speech Synthesizer. We define the reference loss as

$$L_{\text{reference}} = \sum_j \lambda_j \| \widetilde{C}_j^t - C_j^t \|_2^2$$

where weighting parameters λ_j are chosen as follows: voice weight $\lambda_\alpha = 1.8$, loudness $\lambda_L = 1.5$, pitch $\lambda_{f_0} = 0.4$, formant frequencies $\lambda_{f_1} = 3, \lambda_{f_2} = 1.8, \lambda_{f_3} = 1.2, \lambda_{f_4} = 0.9, \lambda_{f_5} = 0.6, \lambda_{f_6} = 0.3$, formant amplitudes $\lambda_{a_1} = 4, \lambda_{a_2} = 2.4, \lambda_{a_3} = 1.2, \lambda_{a_4} = 0.9, \lambda_{a_5} = 0.6, \lambda_{a_6} = 0.3$, broadband filter frequency $\lambda_{f_u} = 10$, amplitude $\lambda_{a_u} = 4$, bandwidth $\lambda_{b_u} = 4$. Similar to speech-to-speech auto-encoding, we add supervision loss for pitch and formant frequencies derived by the Praat method and use the MSS and STOI loss to measure the difference between the reconstructed spectrograms and the ground truth spectrogram. The overall training loss for the ECoG Decoder is:

$$L = L_{\text{MSS}} + \lambda_1 L_{\text{STOI}} + \lambda_2 L_{\text{supervision}} + \lambda_3 L_{\text{reference}}$$

where weighting parameters λ_i are empirically optimized to be $\lambda_1 = 1.2, \lambda_2 = 0.1, \lambda_3 = 1$, through the same parameter search process described for training the Speech Encoder.

We use the Adam optimizer [46] with hyper-parameters: $lr = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$ to train both the auto-encoder (including the Speech Encoding and Speech Synthesizer) and the ECoG Decoder. We train a separate set of models for each participant. As mentioned earlier, we randomly selected 50 out of 400 trials per participant as the test data and used the rest for training.

3.3.5 Contribution Analysis Using the Occlusion Method

To measure the contribution of the cortex region under each electrode to the decoding performance, we adopt an occlusion-based method that calculates the change in the PCC between the decoded and the ground-truth spectrograms when an electrode signal is occluded (i.e., set to zeros), as in [39]. This method enables us to reveal the critical brain regions for speech production. We use the following notations: $\mathrm{S}^t(f)$: the ground truth spectrogram; $\hat{\mathrm{S}}^t(f)$: the decoded spectrogram with "intact" input (i.e., all ECoG signals are used); $\hat{\mathrm{S}}_i^t(f)$: the decoded spectrogram with the i-th ECoG electrode signal being occluded; $r(\cdot,\cdot)$: correlation coefficient between two signals. The contribution of i-th electrode for a particular participant is defined as

$$\mathrm{C}^i = \mathrm{Mean}\left\{r\left(\mathrm{S}^t(f), \hat{\mathrm{S}}^t(f)\right) - r\left(\mathrm{S}^t(f), \hat{\mathrm{S}}_i^t(f)\right)\right\}$$

where $\mathrm{Mean}\{\cdot\}$ denotes averaging across all testing trials of the participant.

We generate the contribution map on the standardized Montreal Neurological Institute (MNI) brain anatomical map by diffusing the contribution of each electrode of each participant (with a corresponding location in the MNI coordinate) into the adjacent area within the same anatomical region using a Gaussian kernel and then averaging the resulting map from all participants. To account for the non-uniform density of the electrodes in different regions and across the

participants, we normalize the sum of the diffused contribution from all the electrodes at each brain location by the total number of electrodes in the region across all participants.

We estimate the noise level for the contribution map to assess the significance of our contribution analysis. To derive the noise level, we trained a shuffled model for each participant by randomly pairing the mismatched speech segment and ECoG segment in the training set. We derive the average contribution map from the shuffled models for all participants using the same occlusion analysis described earlier. The resulting contribution map is used as the noise level. Contribution levels below the noise levels at corresponding cortex locations are assigned a value of 0 (white) in Fig. 3.9.

3.4 Results

3.4.1 Experiments Design

We collected neural data from 48 native English-speaking participants (26 female, 22 male) with refractory epilepsy who had electrocorticographic (ECoG) subdural electrode grids implanted at NYU Langone Hospital. Five participants had hybrid-density (HB) sampling, and 43 had low-density (LD) sampling. The ECoG array was implanted on the left hemisphere for 32 participants and on the right for 16. The Institutional Review Board of NYU Grossman School of Medicine approved all experimental procedures. After consulting with the clinical care provider, a research team member obtained written and oral consent from each participant. Each participant performed five tasks [47] to produce target words in response to auditory or visual stimuli. The tasks were Auditory Repetition (AR, repeating auditory words), Auditory Naming (AN, naming a

word based on an auditory definition), Sentence Completion (SC, completing the last word of an auditory sentence), Visual Reading (VR, reading aloud written words), and Picture Naming (PN, naming a word based on a color drawing).

For each task, we used the exact 50 target words with different stimulus modalities (auditory, visual, etc.). Each word appeared once in the AN and SC tasks and twice in the others. The five tasks involved 400 trials with corresponding word production and ECoG recording for each participant. The average duration of the produced speech in each trial was 500 ms.

3.4.2 Data collection and preprocessing

The study recorded ECoG signals from the perisylvian cortex (including STG, IFG, pre-central, and postcentral gyri) of 48 participants while they performed five speech tasks. A microphone recorded subjects' speech and was synchronized to the clinical Neuroworks Quantum Amplifier (Natus Biomedical, Appleton, WI), which captured ECoG signals. The ECoG array consisted of 64 standard 8×8 macro contacts (10 mm spacing) for 43 participants with low-density sampling. For five participants with hybrid-density sampling, the ECoG array also included 64 additional interspersed smaller electrodes (1 mm) between the macro contacts (providing 10 mm center-to-center spacing between macro contacts and 5 mm center-to-center spacing between micro/macro contacts; PMT corporation, Chanassen, MN). See Fig. 3.7b. This FDA-approved array was manufactured for this study. A research team member informed participants that the additional contacts were for research purposes during consent. Clinical care solely determined placement location across participants (32 left hemispheres; 16 right hemispheres). The decoding models were trained separately for each participant using all trials except 10 randomly selected ones from each

task, leading to 350 trials for training and 50 for testing. The reported results are for testing data only.

We sampled ECoG signals from each electrode at 2048 Hz and downsampled them to 512 Hz before processing. The electrodes with artifacts (e.g., line noise, poor contact with the cortex, high amplitude shifts) were rejected. The electrodes with interictal and epileptiform activity were also excluded from the analysis. The mean of a common average reference (across all remaining valid electrodes and time) was subtracted from each individual electrode. After the subtraction, the Hilbert transform extracted the envelope of the high gamma (70-150 Hz) component from the raw signal, which is then down-sampled to 125 Hz. A reference signal was obtained by extracting a silent period of 250 ms before each trial's stimulus period within the training set and averaging the signals over these silent periods. Each electrode's signal was normalized to the reference mean and variance (i.e., z-score). The data preprocessing pipeline is coded in Matlab and Python.

For participants with noisy speech recordings, we applied spectral gating to remove stationary noise from the speech using an open-source tool [48].

To pre-train the auto-encoder, including the Speech Encoder and Speech Synthesizer, unlike our prior work in [39], which completely relied on unsupervised training, we provide supervision for some speech parameters to improve their estimation accuracy further. Specifically, we use the Praat method [45] to estimate pitch and four formant frequencies ($f^t_{v,\text{i=1 to 4}}$, in Hz) from the speech waveform. The estimated pitch and formant frequency are resampled to 125 Hz, the same as the ECoG signal and spectrogram sampling frequency. The mean square error between these speech parameters generated by the Speech Encoder and the ones estimated by the Praat method is used as a supervised reference loss, in addition to the unsupervised spectrogram reconstruction and

STOI losses, making the training of the auto-encoder semi-supervised.

We employed our speech decoding framework across N=48 participants who consented to complete a series of speech tasks (see Methods Section: Experiments Design). The participants were undergoing treatment for refractory Epilepsy with implanted electrodes for their clinical care. During the hospital stay, we acquired synchronized neural and acoustic speech data. ECoG data were obtained from five participants with hybrid-density(HB) sampling (clinical-research grid) and 43 participants with low-density(LD) sampling (standard clinical grid), who took part in five speech tasks: Auditory Repetition (AR), Auditory Naming (AN), Sentence Completion (SC), Word Reading (WR), and Picture Naming (PN). These tasks were designed to elicit the same set of spoken words across tasks while varying the stimulus modality. We provided 50 repeated unique words (400 total trials per participant), all of which were analyzed locked to the onset of speech production. We trained a model for each participant using 80% of available data for this participant and evaluated the model on the remaining 20% of data (with the exception of the more stringent word-level cross-validation).

3.4.3 Speech Decoding Performance and Causality

We first aimed to directly compare decoding performance across different architectures, including those that have been employed in the neural speech decoding literature (recurrent and convolutional) and transformer-based models. While any decoder architecture could be used for the ECoG Decoder in our framework employing the same Speech Encoder guidance and Speech Synthesizer, we focused on three representative models for convolution (ResNet), recurrent (LSTM), and transformer (SWIN). Note that either model can be configured to use temporally non-causal or causal operations. Our results show that ResNet

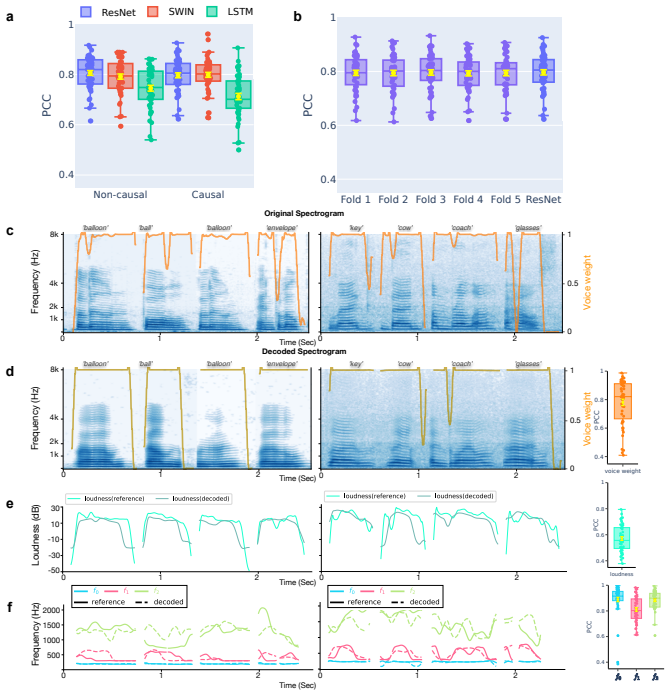


FIGURE 3.4: | **Decoding performance comparing the original and decoded spectrograms across non-causal and causal models. a,** Performance of ResNet, SWIN, and LSTM models with non-causal and causal operations. The PCC between the original and decoded spectrogram is evaluated on the held-out testing set and shown for each participant. Each data point corresponds to a participant's average PCC across testing trials. **b,** A stringent cross-validation showing the causal ResNet model performance on unseen words during training from 5-folds where we ensured the training and validation sets in each fold do not overlap in unique words. The performance across all five validation folds is comparable to our trial-based validation, denoted for comparison as ResNet (identical to the ResNet causal model in (a). **c-f,** Example of decoded spectrograms and speech parameters by the causal ResNet model for eight words (from two participants) and the PCC values between the decoded and reference speech parameters across all participants. Spectrograms of the original (**c**) and decoded (**d**) speech are shown with orange curves overlaid representing the reference voice weight learned by the Speech Encoder (**c**) and the decoded voice weight by the ECoG Decoder (**d**). The PCC between the decoded and reference voice weights is shown on the right across all participants. **e,** The decoded and reference loudness parameters are shown for the eight words, and the PCC of the decoded loudness parameters across participants (boxplot on the right). **f,** Decoded (dashed) and reference (solid) parameters for pitch (f_0) and the first two formants (f_1 and f_2) are shown for the eight words as well as the PCC across participants (boxplots to the right). All box plots depict the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers) across all participants (N=48), and the yellow error bars denote the mean $\pm$ standard error of the mean (SEM) across participants.

outperforms the other models, providing the highest Pearson correlation coefficient (PCC) across N=48 participants (mean PCC=0.806, 0.797 for non-causal, causal respectively) but closely followed by SWIN (mean PCC=0.792, 0.798 for non-causal, causal respectively) shown in Fig. 3.4**a**. We find the same conclusion when evaluating three models using STOI+ [28], shown in Fig. 3.5**a**. The causality of machine learning models for speech production has important implications for BCI applications. A causal model only uses past and current neural signals to generate speech. In contrast, non-causal models use past, present, and future neural signals. Previous reports typically employed non-causal models [5,8,10,25], which could potentially use neural signals related to auditory and speech feedback that is unavailable in real-time applications. Optimally, only the causal direction should be employed. Therefore, we compared the performance of the same models with non-causal or causal temporal operations. Fig. 3.4**a** compares the decoding results of our models' causal and non-causal versions. The causal ResNet model (PCC=0.797) achieved performance comparable to that of the non-causal model (PCC=0.806) with no significant differences between the two (Wilcoxon two-sided signed-rank test $p = 0.093$). The same was true for the causal SWIN model (PCC=0.798) and its non-causal (PCC=0.792) counterpart (Wilcoxon two-sided signed-rank test $p = 0.196$). In contrast, the performance of the causal LSTM model (PCC=0.712) was significantly inferior to that of its non-causal (PCC=0.745) version (Wilcoxon two-sided signed-rank test $p = 0.009$). Further, the LSTM model showed consistently lower performance than ResNet and SWIN. However, we did not find significant differences between causal ResNet and causal SWIN performance (Wilcoxon two-sided signed-rank test $p = 0.587$). Since the ResNet and SWIN models had the highest performance and were on par with each other and their causal counterparts, we chose to focus further analyses on these causal models, which we

believe are best suited for prosthetic applications.

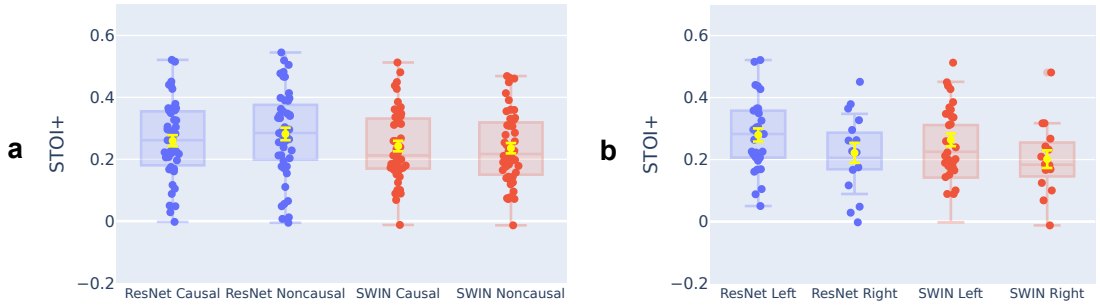


FIGURE 3.5: | **Comparison of decoding STOI+ under different settings of the 3D ResNet and 3D SWIN models. a,** Performance of ResNet and SWIN models with non-causal and causal operations across all participants (N=48; 43 low-density ECoG grids and 5 hybrid density grids). The STOI+ between the original and decoded spectrogram is evaluated on the held-out testing set and shown for each participant. Each data point corresponds to a participant's average PCC across all testing trials. The boxplot represents the median, 25th and 75th quantiles across participants, and the yellow error bar denotes the mean and standard error of the mean. As with PCC (Fig. 2a in the manuscript), ResNet and SWIN models perform similarly, but the LSTM model is significantly worse. **b,** STOI+ comparison between left and right hemisphere participants, using causal ResNet and SWIN models. No statistically significant decoding performance differences exist between left (N=32) and right (N=16) hemisphere participants (ResNet independent t-test, p=0.166; SWIN independent t-test, p=0.114), although the left hemisphere participants have slightly greater mean STOI+. All box plots depict the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers). The yellow error bars denote the mean $\pm$ SEM.

To ensure our framework can generalize well to unseen words, we added a more stringent word level cross-validation wherein random (10 unique) words were held entirely out during training (including both pre-training of Speech Encoder and Speech Synthesizer and training of ECoG Decoder). This ensures that different trials from the same word could not appear in both the training and testing sets. Results shown in Fig. 3.4**b** demonstrate that performance on the held-out words is comparable to our standard trial-based held-out approach

(i.e., Fig. 3.4**a**. "ResNet"). It is encouraging that the model can decode unseen validation words well, regardless of which words were held out during the training.

Next, we show the performance of the ResNet causal decoder on the level of single words across two representative participants (low-density grids). The decoded spectrograms accurately preserve the spectro-temporal structure of the original speech (Fig. 3.4**c,d**). We also compare the decoded speech parameters with the reference parameters. For each parameter, we calculate the PCC between the decoded time series and the reference sequence showing an average PCC of 0.781 (voice weight, Fig. 3.4**d**), 0.571 (loudness, Fig. 3.4**e**), 0.889 (pitch, f_0, Fig. 3.4**f**), 0.812 (first formant f_1, Fig. 3.4**f**), 0.883 (second formant f_2, Fig. 3.4**f**). The accurate reconstruction of the speech parameters, especially the pitch, voice weight, and first two formants, is essential for accurate speech decoding and naturalistic reconstruction that mimics the participant's voice. We also provide a non-causal version of Fig. 3.4 in Fig. 3.6. The fact that both non-causal and causal models could yield reasonable decoding results is encouraging.

Left vs Right Hemisphere Decoding and Effect of Electrode Density

Most speech decoding studies focused on the language and speech-dominant left hemisphere ([49]). However, little is known about decoding speech representations from the right hemisphere. To this end, we compared left vs right hemisphere decoding performance across our participants to establish the feasibility of a right hemisphere speech prosthetic. For both our ResNet and SWIN decoders, we found robust speech decoding from the right hemisphere (ResNet PCC=0.790, SWIN PCC=0.798), which were not significantly different from the left (Fig. 3.7**a**, ResNet independent t-test, p=0.623; SWIN independent t-test,

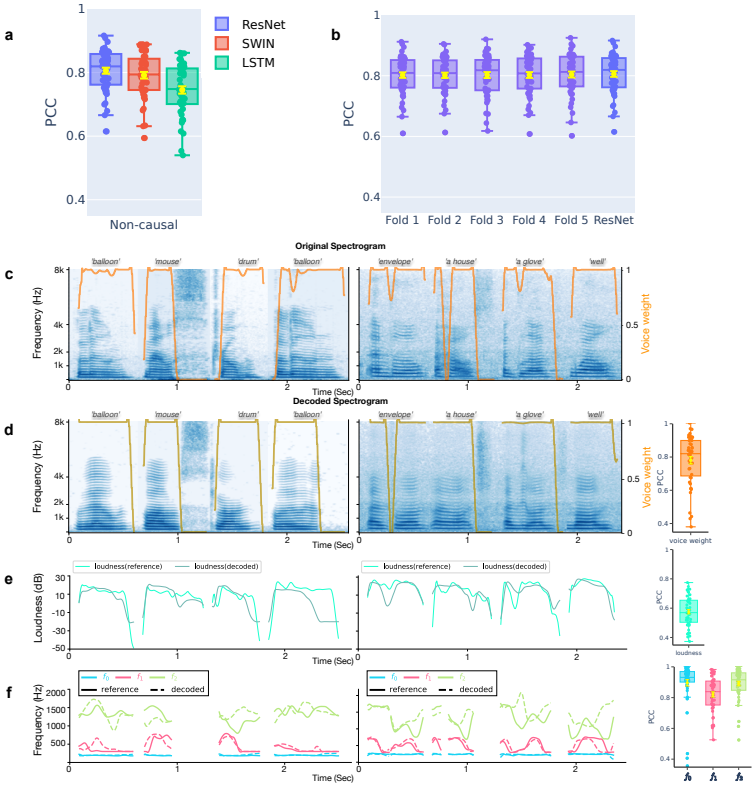


FIGURE 3.6: | **Decoding performance comparing the original and decoded spectrograms across non-causal models.** **a,** PCC between the original and decoded spectrograms by non-causal ResNet, 3D SWIN, and LSTM models for all participants (N=48), including 43 participants with LD ECoG grids and 5 participants with HB grids, evaluated on the held-out testing set, which contains different speech trials from the training set but includes overlapping words. **b,** ResNet model PCC on unseen words during training from a 5-fold cross-validation study where the training and validation sets in each fold have non-overlapping words. The performance across five folds is similar to randomly selected test trials. **c-f,** Example decoded spectrograms and speech parameters by the non-causal ResNet model for four words each from two participants and PCC between the decoded and reference speech parameters across all participant trials. **c,d,** Comparison of original (**c**) and decoded (**d**) spectrograms. The orange curves overlaid on the spectrograms in **c** and **d** show the reference voice weight generated by the speech encoder and the decoded voice weight by the ECoG decoder, respectively. The box plot in **d** shows the PCC between the decoded and reference voice weight for all participants (N=48). **e,** Decoded loudness parameter compared to reference loudness parameter. The box plot shows the PCC of the decoded and reference loudness parameters (N=48). **f,** Comparison of the decoded and reference f_0 and f_1, f_2 (N=48). The results achieved with non-causal models follow a very similar trend as those obtained with causal models, shown in Fig. 3.4f. All box plots depict the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers). The yellow error bars denote the mean $\pm$ SEM.

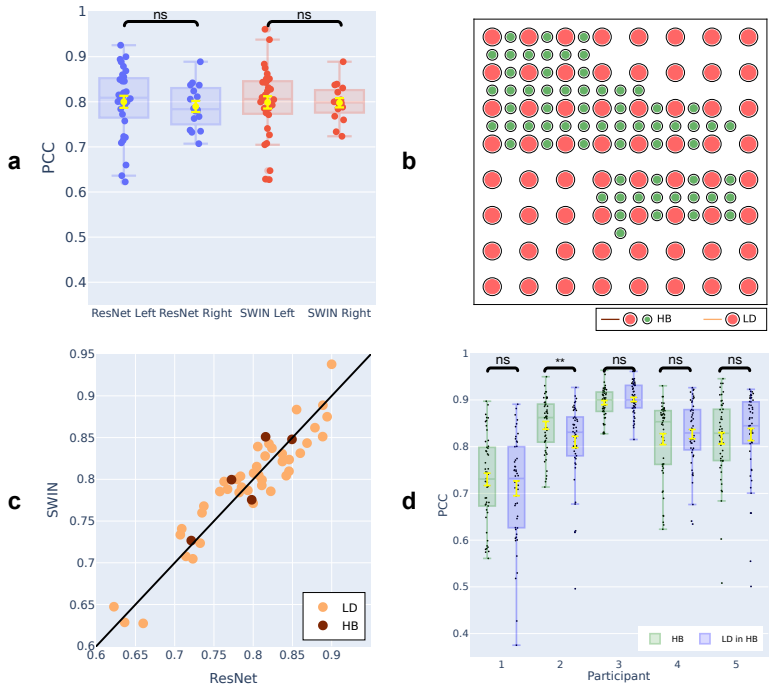


FIGURE 3.7: | **Comparison of decoding PCC under different settings of the 3D ResNet and 3D SWIN models.** **a,** Comparison between left and right hemisphere participants, using causal models. No statistically significant differences (ResNet independent t-test, p=0.623; SWIN Wilcoxon independent t-test, p=0.968) in PCC exist between left (N=32) and right (N=16) hemisphere participants. **b,** An example hybrid density ECoG array with a total of 128 electrodes. The 64 electrodes marked in red correspond to a low-density placement. The remaining 64 green electrodes, combined with red electrodes, reflect a hybrid density placement. **c,** Comparison between causal ResNet and causal SWIN models for the same participant across participants with hybrid-density (HB, N=5) or low-density (LD, N=43) ECoG grids. The two models have similar decoding performances from the HB grids and LD grids. **d,** The decoding PCC across 50 test trials by the ResNet model for HB (N=5) participants when all electrodes are used vs when only LD-in-HB electrodes (N=5) are considered. There are no statistically significant differences for 4 out of 5 participants (Wilcoxon two-sided signed-rank test, $p = 0.114, 0.003, 0.0773, 0.472, 0.605$ respectively). All box plots depict the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers). The yellow error bars denote the mean $\pm$ SEM. Distributions were compared with each other as indicated using the Wilcoxon two-sided signed-rank test and independent t-test. $**P < 0.01$.

p=0.968). A similar conclusion holds when evaluating STOI+ (Fig. 3.5**b**, ResNet independent t-test, p=0.166; SWIN independent t-test, p=0.114). While these results suggest that it may be feasible to use neural signals in the right hemisphere to decode speech for patients who suffer damage to the left hemisphere and are unable to speak ([50]), it remains unknown if intact left hemisphere cortex is necessary to allow for speech decoding from the right hemisphere until tested in such patients.

3.4.4 Effect of Electrode Density

Next, we assessed the impact of the electrode sampling density for speech decoding, as many previous reports employ higher-density grids (0.4 mm) with more closely spaced contacts than typical clinical grids (1 cm). Five participants consented to hybrid grids (i.e., HB, see Fig. 3.7**b**), which had typical low density (i.e., LD) electrode sampling as well as additional electrodes interleaved. The hybrid density grids provided a decoding performance similar to clinical low-density grids in terms of PCC (Fig. 3.7**c**), with a slight advantage in STOI+, shown in Fig. 3.8**b**. To ascertain if the additional spatial sampling indeed provides improved speech decoding, we compared models that decode speech based on all the hybrid electrodes vs. only the low-density electrodes in patients with HB grids (comparable to our other LD participants). Our findings (Fig. 3.7**d**) suggest that decoding results were not significantly different from each other (with the exception of Participant 2) in terms of both PCC and STOI+ (Fig. S3**c**). Together, these results suggest that our models can learn speech representations well from both the high and low spatial sampling of the cortex with the exciting finding of robust speech decoding from the right hemisphere.

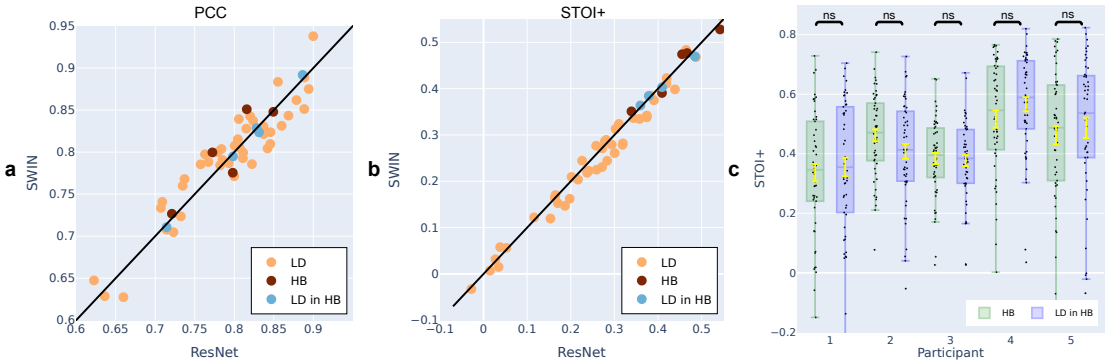


FIGURE 3.8: | **Comparison of decoding PCC and STOI+ of the causal 3D ResNet and 3D SWIN models for the same participant across participants with hybrid-density (HB, N=5), low-density (LD, N=43), and only LD-in-HB ECoG grids.** In terms of PCC (**a**), both models have similar decoding performances on both HB, LD, and LD-in-HB participants. In terms of STOI+ (**b**), both models have slightly better performance on the HB participants, and the ResNet has similar performance as the SWIN model for both HB and LD participants. The LD-in-HB (N=5) and HB participants (N=5) have very similar performances in both PCC and STOI+ (**c**). The decoding STOI+ by the ResNet model for HB participants (N=5) when all electrodes are used vs when only LD-in-HB electrodes (N=5) are considered. There are no statistically significant differences for all participants (Wilcoxon two-sided signed-rank test, p-value = 0.626, 0.146, 0.881, 0.058, 0.414). In 3 out of 5 participants, using LD-in-HB electrodes even gives us higher STOI+ compared with using HB only. Box plot depicts the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers). The yellow error bars denote the mean $\pm$ SEM. Distributions were compared with each other as indicated using the Wilcoxon two-sided signed-rank test. $ns > 0.05$.

Lastly, we investigate what cortical regions contribute to decoding, providing insight for targeted implantation in future prosthetics, especially on the right hemisphere, which has not been studied. We employed an occlusion approach to quantify the contributions of different cortical sites to speech decoding. If a region is involved in decoding, occluding the neural signal in the corresponding electrode (i.e., setting the signal to zero) would reduce the accuracy (PCC) of the reconstructed speech on the testing data (see Methods Section: Contribution Analysis). Therefore, we measure each region's contribution by decoding PCC reduction when the corresponding electrode is occluded. We analyzed all electrodes and participants for causal and non-causal ResNet and SWIN decoder versions. The results in Fig. 3.9 show similar contribution values for both ResNet and SWIN models (see Fig. S8, S9 for noise level contribution). The Non-Causal models show enhanced auditory cortex contributions compared with the Causal models, implicating auditory feedback in decoding and underlying the importance of employing only Causal models during speech decoding because neural feedback signals are not available for real-time decoding applications. Further, across the Causal models, both the right and left hemispheres show similar contributions across the sensorimotor cortex, especially on the ventral portion, which provides possible feasibility for right hemisphere neural prosthetics.

3.4.5 Contribution analysis

3.5 Discussion

Our novel pipeline can decode speech from neural signals leveraging interchangeable architectures for the ECoG Decoder and a novel differentiable Speech

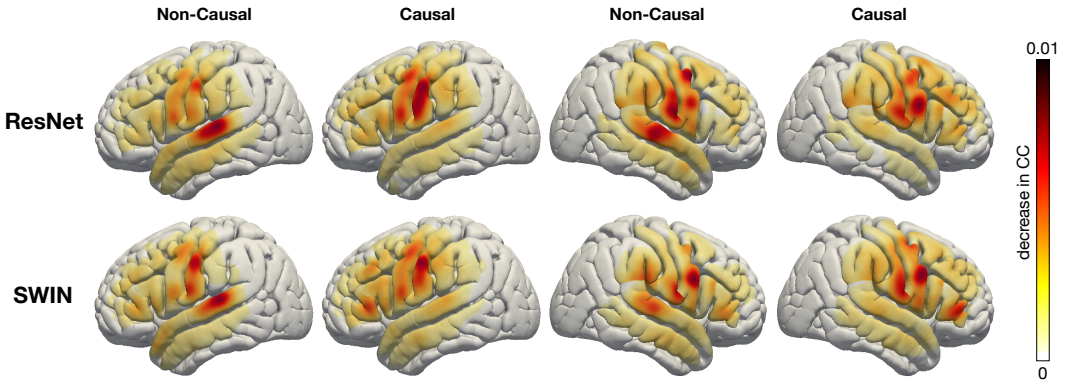


FIGURE 3.9: | We visualize the contribution of each cortical location to the decoding result by both causal or non-causal decoding models through an occlusion analysis. The contribution of each electrode region in each participant is projected onto the standardized Montreal Neurological Institute (MNI) brain anatomical map and then averaged over all participants. Each subplot shows the causal or non-causal contribution of different cortical locations (red indicates higher contribution while yellow indicates lower contribution). For visualization purposes, we normalize the contribution of each electrode location by the local grid density since we have multiple participants with non-uniform density.

Synthesizer. Our training process relies on estimating guidance speech parameters from the participants' speech using a pre-trained Speech Encoder. This strategy enabled us to train ECoG Decoders with limited corresponding speech and neural data, which can produce natural-sounding speech when paired with our Speech Synthesizer. Our approach was highly reproducible across participants (N=48), providing evidence for successful causal decoding with convolutional (ResNet) and transformer (SWIN) architectures, both outperforming recurrent architecture (LSTM). Our framework can successfully decode from both high and low spatial sampling with high levels of decoding performance. Lastly, we provide potential evidence for robust speech decoding from the right hemisphere and the spatial contribution of cortical structures to decoding across the hemispheres.

Our decoding pipeline showed robust speech decoding across participants,

leading to a Pearson correlation coefficient within the range of 0.62-0.92 (Fig. 3.4**a**, Causal ResNet mean 0.797, median 0.805) between decoded and the ground truth speech across several architectures. We attribute our stable training and accurate decoding to the carefully designed components of our pipeline (e.g., Speech Synthesizer, Speech Parameter Guidance) and multiple improvements (see Methods Section: Speech Synthesizer, ECoG Decoder, and Model Training) over our previous approach on the subset of participants with hybrid density grids [39]. Previous reports investigated speech or text decoding used linear models [22, 23, 51], transitional probability [4, 52], recurrent neural networks [5, 10, 14, 25], convolutional neural networks [8, 39], and other hybrid or selection approaches [9, 15, 16, 24, 26]. Overall, our results are similar to (or better than) many previous reports (54% of our participants showed higher than 0.8 decoding PCC, see Fig. 3.7**c**). However, a direct comparison is complicated by multiple factors. Previous reports vary on reported performance metrics, as well as the stimuli decoded (e.g., continuous speech vs. single words) and the cortical sampling (i.e., high vs. low density, depth electrodes compared with surface grids). Our publicly available pipeline, which can be used across multiple neural network architectures and tested on various performance metrics, can facilitate the research community to conduct more direct comparisons while still adhering to a high accuracy of speech decoding.

The temporal causality of decoding operations, critical for real-time BCI applications, has not been considered by most prior studies. Many of these non-causal models relied on auditory (and somatosensory) feedback signals. Our analyses show that non-causal models rely on a robust contribution from STG,

which is mostly eliminated using a causal model (Fig. 3.9). We believe that non-causal models would show limited generalizability for real-time BCI applications due to their over-reliance on feedback signals that may be absent (if no delay is allowed) or incorrect (if a short latency is allowed during real-time decoding). Some approaches used imagined speech, which avoids feedback during training [24], or showed generalizability to mimed production lacking auditory feedback [14,25]. However, most reports still employ non-causal models, which cannot rule out feedback during training and inference. Indeed, our contribution maps show robust auditory cortex recruitment for non-causal ResNet and SWIN models (Fig. 3.9), in contrast to the causal counterparts, which decode based on more frontal regions. Further, recurrent neural networks widely used in the literature [5, 14] are typically bi-directional, producing non-causal behaviors and longer latencies for prediction during real-time application. However, the uni-directional causal results are typically not reported. The recurrent network we tested performed the worst when trained with one direction (Causal LSTM, Fig. 3.4a). While our current focus was not real-time decoding, we are able to synthesize speech from neural signals with under 50 ms delay, which provides minimal auditory delay interference allowing for normal speech production [53, 54]. Our data suggest that causal convolutional and transformer models can perform on par with their non-causal counterparts and recruit more relevant cortical structures for real-time decoding.

In our study, we leveraged an intermediate speech parameter space together with a novel differentiable Speech Synthesizer to decode subject-specific naturalistic speech (Fig. 3.1). Previous reports used varying approaches to model speech, including an intermediate kinematic space [25], an acoustically relevant intermediate space using HuBERT features [14] derived from self-supervised speech reconstruction task [17], an intermediate random vector (i.e., GAN) [11],

or direct spectrogram representations [8,25,55,56]. Our choice of speech parameters as the intermediate representation allowed us to decode subject-specific acoustics. Our intermediate acoustic representation led to significantly more accurate speech decoding than directly mapping ECoG to the speech spectrogram [57], and than mapping ECoG to a random vector, which is then fed to a GAN-based speech synthesizer [11].

Unlike the kinematic representation, our acoustic intermediate representation using speech parameters and the associated speech synthesizer enables our decoding pipeline to produce natural-sounding speech that preserves patient-specific characteristics, which would be lost with the kinematic representation.

Our Speech Synthesizer is motivated by classical vocoder models for speech production (generating speech by passing an excitation source, harmonic or noise, through a filter [58, 59] and is fully differentiable, facilitating the training of the ECoG decoder using spectral losses through backpropagation. Further, the guidance speech parameters needed for training the ECoG decoder can be obtained using a Speech Encoder that can be pre-trained without requiring neural data. Thus, it could be trained using older speech recordings or a proxy speaker chosen by the patient in the case of patients without the ability to speak. Training the ECoG decoder using such guidance, however, would require us to revise our current training strategy to overcome the challenge of misalignment between neural signals and speech signals, which is a scope of our future work. Additionally, the low-dimensional acoustic space and pre-trained Speech Encoder (for generating the guidance) using speech signals only alleviate the limited data challenge in training ECoG to speech decoder and provide a highly interpretable latent space. Finally, our decoding pipeline is generalizable to unseen words (Fig. 3.4b). This provides an advantage compared to pattern

matching approaches [26] that produce subject-specific utterances but with limited generalizability.

Many prior studies employed high-density electrode coverage over the cortex, providing many distinct neural signals [5, 10, 25, 51, 56]. One question we directly addressed was whether higher-density coverage improves decoding. Surprisingly, we found a high decoding performance in terms of spectrogram PCC for both low-density and higher (hybrid) density grid coverage (Fig. 3.7**c**). Further, comparing the decoding performance obtained using all electrodes in our hybrid-density participants vs. using only the low-density electrodes in the same participants showed that decoding did not significantly differ (Fig. 3.7**d**, albeit for one participant). We attribute these results to our ECoG decoder's ability to extract speech parameters from neural signals as long as there is sufficient peri-sylvian coverage, even in low-density participants.

A striking result was the robust decoding from right hemisphere cortical structures as well as the clear contribution of the right peri-sylvian cortex. Our results are consistent with the idea that syllable-level speech information is represented bilaterally [60]. However, our findings suggest that speech information is well-represented in the right hemisphere. Our decoding results could directly lead to speech prostheses for patients who suffer from expressive aphasia or apraxia of speech. Some previous studies have shown limited right hemisphere decoding of vowels ([61]) and sentences [62]. However, the results were mostly mixed with left-hemisphere signals. While our decoding results provide evidence for a robust representation of speech in the right hemisphere, it is important to note that these regions are likely not critical for speech, as evidenced by the few studies that have probed both hemispheres using electrical stimulation mapping [63, 64]. Further, it is unclear if the right hemisphere would contain sufficient information for speech decoding if the left hemisphere

is damaged. It would be necessary to collect right-hemisphere neural data from left-hemisphere-damaged patients in order to verify we can still achieve acceptable speech decoding. However, we believe right-hemisphere decoding is still an exciting avenue as a clinical target for patients who are unable to speak due to left-hemisphere cortical damage.

There are several limitations in our study. First, our decoding pipeline requires speech training data paired with ECoG recordings, which may not exist for paralyzed patients. This could be potentially mitigated by using neural recordings during imagined or mimed speech and the corresponding older speech recordings of the patient or speech by a proxy speaker chosen by the patient. As discussed earlier, we would need to revise our training strategy to overcome the temporal misalignment between the neural signal and the speech signal.

Second, our ECoG Decoder models (3D ResNet and 3D SWIN) assume a grid-based electrode sampling, which may not be the case. Future work should develop model architectures that are capable of handling non-grid data, such as strips and depth electrodes (stereo intracranial EEG). Importantly, such decoders could replace our current grid-based ECoG Decoders while still being trained using our overall pipeline. Lastly, our focus in this study was on word-level decoding limited to a vocabulary of 50 words, which may not be directly comparable to sentence-level decoding. Specifically, two recent studies provided robust speech decoding in a few chronic patients implanted with intracranial ECoG [14] or Utah array [65] that leveraged a large amount of data available in one patient in each study. It is noteworthy that these studies employ varying approaches in constraining their neural predictions. Metzger et al. employed a pre-trained large transformer model leveraging directional attention to provide the guidance HuBERT features for their ECoG decoder. In contrast, Willet et al.

decode at the level of phonemes and use transition probability models at both phoneme and word levels to constrain decoding. Our study is much more limited in terms of data. However, we are able to achieve good decoding results across a large cohort of patients through the use of a compact acoustic representation (rather than learnt contextual information). We expect that our approach can help improve generalizability for chronically implanted patients.

To summarize, our neural decoding approach, capable of decoding natural-sounding speech from 48 participants, provides the following major contributions. First, our proposed intermediate representation uses explicit speech parameters and a novel differentiable speech synthesizer, which enables interpretable and acoustically accurate speech decoding. Second, we directly consider the causality of the ECoG Decoder, providing strong support for causal decoding, which is essential for real-time BCI applications. Third, our promising decoding results using low sampling density and right hemisphere electrodes shed light on future neural prosthetic devices using low-density grids and in patients with damage to the left hemisphere. Last but not least, we have made our decoding framework open to the community with documentation (`https://github.com/flinkerlab/neural_speech_decoding`), and we trust that this open platform will help propel the field forward, supporting reproducible science.

3.6 Conclusion

In this chapter, we present a neural speech decoding framework leveraging deep learning and speech synthesis. Our neural decoding approach, capable of decoding natural-sounding speech from 48 participants, provides the following major contributions. First, our proposed latent representation using explicit speech parameters and a differentiable Speech Synthesizer enables interpretable

and intelligible speech decoding. Second, we directly consider the causality of the ECoG Decoder, providing strong support for causal decoding, which is essential for real-time BCI applications. Third, our promising decoding results using low sampling density and right hemisphere electrodes shed light on future neural prosthetic devices in patients with damage to the left hemisphere. Last but not least, we have made our framework open to the community with documentation (`https://xc1490.github.io/nsd/`), and we trust that this open platform will help propel the field forward, supporting reproducible science.

This work is a close collaboration with Dr. Ran Wang. He works with the speech synthesizer. I made modifications to the framework to extend it to many low-density participants. We increased the number of prototype filter parameters M from 20 to 80 to enable more expressive and flexible modeling of formant filter shapes, particularly for male speakers. Additionally, we replaced the use of a fixed, statistically derived background noise spectrum with a learnable noise model, allowing the background noise distribution $B(f)$ to be optimized jointly during auto-encoder training. To help the learning of the speech components. We added two additional losses for model training: the supervision loss of pitch and formants frequencies, and the differentiable STOI+-based loss to boost the intelligibility. We adopt the 3D Swin Transformer model for the ECoG decoder part, which leverages temporal and spatial joint attention to aggregate the features from ECoG signals. We showed that our model has good generalization abilities on 43 low-density participants with ECoG placed on either the left or right hemispheres. We wrote a paper based on the results shown in this chapter, which has been published in Nature Machine Intelligence [20].

Chapter 4

Transformer-based neural speech decoding from surface and depth electrode signals

4.1 Introduction

The decoding pipeline described in Chapter. 3 first applies a Neural Decoder (called ECoG Decoder) to predict time-varying speech parameters and then uses a novel Speech Synthesizer to generate speech spectrograms from speech parameters. Using ResNet [34] or 3D Swin Transformer [66] as the Neural Decoder, high speech decoding performance in terms of PCC between the decoded and ground-truth spectrograms has been achieved. More recent advances in neural-network-based speech decoding are reviewed in Chapter. 2.

The deep neural networks in previous speech decoding studies have architecture designs with several limitations. First, architectures that use spatial convolution among electrodes, e.g., [8] and the 3D ResNet described in Chapter. 3, are only applicable to grid electrodes like an ECoG array and hence do not work with strip or depth electrodes. Vision transformers' absolute position

embeddings and relative positional bias are also based on the 2D or 3D grid index [35, 66–68] and hence are only applicable to grid electrodes ([69, 70] and 3D SWIN described in Chapter. 3). On the other hand, the implantation of depth electrodes (stereotactic EEG or sEEG) has been a more popular neurosurgical approach, which does not require the removal of a large skull portion, with reports of fewer surgical complications [71, 72]. Further, the approach and electrodes employed in sEEG are similar to those used in Deep Brain Stimulation (DBS), which has demonstrated long-term electrode safety, suggesting the possibility of chronic sEEG for speech neuroprostheses [73]. Multiple sEEG depth probes may be implanted, which can assay a wide range of deeper structures and thus may provide additional information not available from the surface of the cortex. Therefore, decoding speech from sEEG signals would have significant clinical advantages.

Secondly, models that use fully connected computations among the electrodes, e.g., [10, 25, 74], can only be trained for a specific participant, as the weights learned depend on the actual locations of the electrodes in the brain and the ordering of the electrodes. Because electrode placement varies quite widely across patients, fully connected architectures cannot be trained effectively with data from multiple participants unless subject-specific layers are introduced to map the original electrode data from different participants to a common feature domain. Likewise, convolutional [8,20,39] models or transformers [20,39,69,70] that leverage grid indices for position embedding cannot generalize well to different participants because they do not specifically consider the locations of the electrodes on the brain. Therefore, studies to date have developed subject-specific models, which suffer from small data challenges as they cannot leverage signals from multiple subjects. Several studies have proposed models that have subject-specific layers along with a shared module that can be trained with data

from multiple participants [9,10,75]. However, such models still require collecting training data for each participant to refine the subject-specific layer, limiting its practical applicability.

In this chapter, we develop a novel transformer-based Neural Decoder that does not rely on a regular grid structure. We call it the Swin transformer with temporal windowing (SwinTW). Instead of relying on the grid index, the model leverages the anatomical location of electrodes in the standardized brain template to learn the attention between electrodes. The proposed Neural Decoder performed better than ResNet and 3D Swin Transformer across 43 participants, given the same grid electrodes, reported in Chapter. 3. The model demonstrated further performance increase by leveraging the off-grid electrodes that cannot be utilized in the previous studies. Importantly, the model demonstrated promising performance given sEEG electrodes across 9 participants. Most significantly, the SwinTW model can be effectively trained with data from multiple participants without any subject-specific layers, and the resulting model can generalize well to participants outside the training cohort.

4.2 Method

4.2.1 Speech Decoding Framework

Our neural decoding framework is trained by following a 2-step approach described in Chapter. 3, shown in Fig. 4.1. In the first step of Speech-to-Speech training, a Speech Encoder is used to extract speech parameters at every time frame (e.g., pitch, formant frequencies, loudness) from the input speech spectrogram, and a differentiable Speech Synthesizer is designed to reconstruct the spectrogram from the speech parameters. The Speech Encoder and the Speech

Synthesizer are trained to match the reconstructed spectrogram with the ground truth. In the second step of Neural-to-Speech training, the Neural Decoder is trained to predict the time-varying speech parameters from neural signals using the speech parameters generated by the Speech Encoder as guidance. The predicted speech parameters from the Neural Decoder are fed to the trained Speech Synthesizer from step 1 to generate the decoded speech spectrogram, which is then converted to the decoded speech waveform.

Following the design from our previous study described in Chapter. 3, the Speech Encoder extracts 18 speech parameters at each time step from the original speech spectrogram, which is then fed to the Speech Synthesizer to reconstruct the original speech spectrogram. The Speech Encoder adopts a simple network architecture with an MLP (Multilayer Perceptron) and temporal convolution. The differentiable speech synthesizer enables the end-to-end training of the speech-to-speech auto-encoding task (see right of Fig. 4.1). Details about the Speech Encoder and Speech Synthesizer and their pretraining using speech signal only can be found in Chapter. 3.

For Neural-to-Speech training, the Neural Decoder first maps neural activity from all input electrodes to a latent feature, which is then used to predict the 18 speech parameters for each time frame, supervised by the speech parameters generated by the Speech Encoder. Then, the speech parameters predicted by the Neural Decoder will be fed into the Speech Synthesizer to generate the decoded spectrogram, which is then converted to the ECoG-decoded speech signal.

At the inference time, only the trained Neural Decoder and the Speech Synthesizer are needed (Fig. 4.1.b).

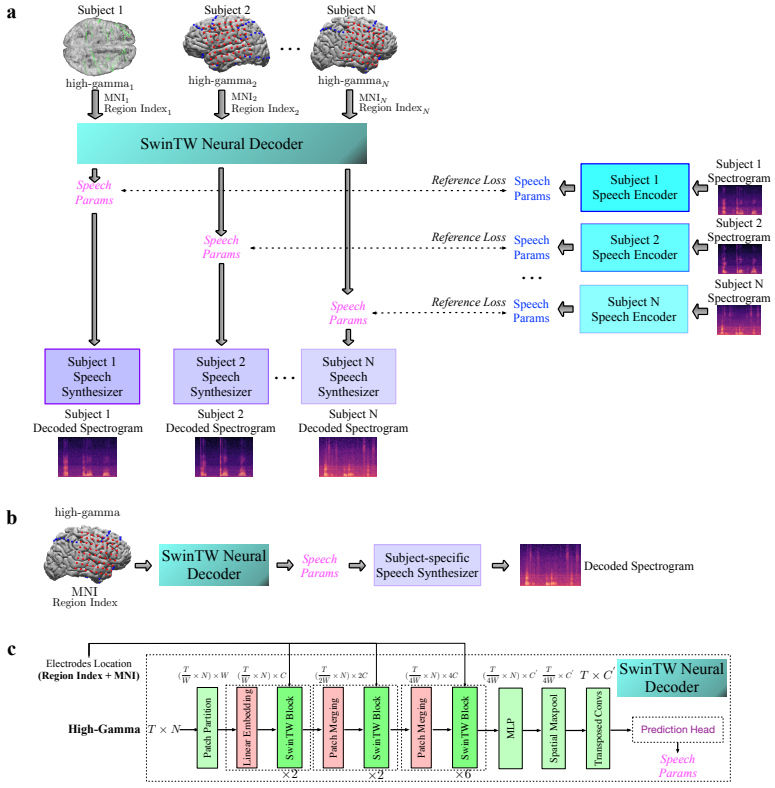


FIGURE 4.1: **a** Multiple-subject Neural Decoder training pipeline. Each participant's neural signal and electrodes' location information (MNI coordinates and ROI index) are fed to a shared SwinTW Neural Decoder to predict speech parameters. The predicted speech parameters are supervised by the speech parameters generated by the subject-specific Speech Encoder from the ground-truth speech spectrogram. Each participant's predicted speech parameters are fed into the corresponding subject-specific Speech Synthesizer to generate a speech spectrogram. Once the shared SwinTW decoder is trained, it can be used to decode the speech from the neural signal of any participant. Note that the same training pipeline can be used to train a subject-specific model using data from a single participant. **b** The inference pipeline: The trained SwinTW decoder takes the neural signals and location information of the electrodes of a participant as the input and generates speech parameters. The speech synthesizer (pre-trained using the speech signal of the participant) then converts the generated speech parameters to the decoded spectrogram. **c** The SwinTW decoder architecture uses three stages of transformer blocks with spatial-temporal attention, with temporal windowing and patch merging to extract features. Transposed temporal convolution is then used to upsample the features back to the temporal dimension of the input. A prediction head module finally generates speech parameters from the upsampled features.

4.2.2 Neural Decoder based on Temporal Swin Transformer

In this Chapter, we develop a novel architecture for decoding speech parameters from electrode signals that do not require electrodes to be on a 2D grid. We name the proposed Neural Decoder a Swin transformer with temporal windowing (SwinTW), inspired by the Swin Transformer [35, 68]. In the vanilla Vision Transformer (ViT) for an image [67], the self-attention layer computes global attention among all tokens (with each token corresponding to an image patch). This global attention causes the absence of the inductive bias of locality and heavy quadratic computational complexity to the input image size. The Swin Transformer solves the problems by grouping tokens into local windows and computing local attention within each window at each self-attention layer. To allow inter-window information exchange, the Swin Transformer shifts the window partition between every two windowed self-attention layers, which prevents different windows from being segregated (details can be found in [35,68]). However, since the Swin Transformer was designed for 2D images (later extended to 3D videos [66]), its architecture assumes that the input is in the format of 2D or 3D grids. Our previous transformer-based Neural Decoder used 3D Swin described in Chapter. 3 which is inspired by [66], where each 3D window includes nearby 2×2 electrodes in two adjacent time steps. In our proposed SwinTW, we made several modifications to allow speech decoding based on electrodes in any topological layout. We still have spatial and temporal attention, but windowing is only applied in the temporal direction to constrain temporal attention. Instead of using electrode location on the 2D grid for spatial positioning information in Chapter. 3, we use the anatomic location of each grid on the cortex (MNI coordinate and brain region index). The architecture of the SwinTW is shown in Fig. 4.1.

Temporal patch partition: In the Swin Transformer [35,66,68] or ViT [67], the input images or videos are partitioned into 2D or 3D patches, and each patch is then mapped to a token with a patch embedding layer. This patch partition requires ordering all the electrodes into a 2D grid and makes the trained model not invariant to the electrode order. To solve this problem, our proposed SwinTW generates tokens from each electrode individually and only partitions the temporal dimension. As shown in Fig. 4.1, given an ECoG signal with the shape of $T \times N$ (T: number of frames, N: number of electrodes), for each electrode, the SwinTW partitions the temporal sequence of neural activity into $\frac{T}{W}$ patches with patch size W. The temporal patch partition generates $\frac{T}{W} \times N$ patches in total, and a linear patch embedding layer is applied to each patch to generate $\frac{T}{W} \times N$ tokens with the latent dimension of C.

Temporal window attention: In Swin transformer [35,66,68], tokens are partitioned into windows, where each window contains a local subset of adjacent tokens, and attention is calculated only among tokens within the same window. In conventional 3D Swin, the windowing is applied spatially and temporally, making the model only suitable for electrodes arranged in a 2D grid. In SwinTW, to remove this grid input constraint, the model only partitions tokens into local windows in the temporal dimension and allows spatial attention across all electrodes (this can be thought of as using a spatial window size that includes all electrodes). Given $N = N_t \times N_s$ tokens (N: total number of tokens, N_t: number of tokens in the temporal dimension, N_s: number of tokens in the spatial dimension, equal to number of electrodes) and window size W_t, the N tokens are partitioned into $\frac{N_t}{W_t}$ windows and attention is calculated among $W_t \times N_s$ tokens within each window.

Temporal patch merging: The Swin Transformer leverages patch merging to achieve inductive bias of locality and hierarchical feature maps. However,

merging nearby patches in the spatial dimension is not feasible when the electrodes are not arranged in a grid. Therefore, instead of using the spatiotemporal patch merging in the 3D Swin Transformer [66], the SwinTW conducts temporal patch merging for each electrode individually. For each electrode, every two consecutive tokens in the temporal dimension with feature dimension C will be concatenated as a $2C$ dimensional latent and get mapped to a $2C$ dimensional merged token.

Grid-free positional embedding: The SwinTW follows Swin Transformers [68] to exploit positional information through relative positional bias. However, instead of using the 2D or 3D grid index difference as the relative position like the Swin Transformer, our SwinTW defines the relative positional bias based on each token's anatomical location and time-frame index. The positional bias is defined as below:

$$Attention(Q, K, V) = Softmax(SIM(Q, K))V \tag{4.1}$$

$$SIM(q_i, k_j) = \frac{q_i k_j}{|q_i||k_j|}/\tau + B_{i,j} \tag{4.2}$$

$$B_{i,j} = MLP(x_i, y_i, z_i, t_i, x_j, y_j, z_j, t_j, x_i - x_j, y_i - y_j, z_i - z_j, t_i - t_j) + r_i \cdot r_j \tag{4.3}$$

Given $Q, K, V \in R^{N \times C}$ (Q, K, V are query, key and value generated from each token, N is number of tokens and C is the latent dimension), shown in equation 4.1, the softmax of $SIM(Q, K)$ for all pairs of token in the window is used to aggregated V (values of tokens within the window) to get the output token values. We define query-key similarity following the scaled cosine attention of SwinV2 [68], defined in equation 4.2. τ is a learnable parameter not shared among attention heads and layers. $B_{i,j}$ is the relative positional bias between token i and token j. In SwinTW, $B_{i,j}$ consists of two terms: MNI-based positional

bias and region-based bias. We project each subject's electrodes to a standardized Montreal Neurological Institute (MNI) brain anatomical map and collect each electrode's x, y, z location in the MNI coordinate. For each token pair, the MNI coordinates of the corresponding electrodes and time-frame index, along with their differences, will be mapped to the MNI-based positional bias with a 2-layer MLP, which is shown in the first term of Eq. (4.3). We also parcellate the standardized brain into regions of interest (ROIs) and learn a dictionary of embeddings for all ROIs, with r_i denoting the embedding features for region i. Given N_r ROIs and N_h attention head, the learnable dictionary has N_h sets of $N_r \times C_r$ region embeddings (C_r is the region embedding dimension). The region embeddings $r_i, \forall i$ are learned during the training. For a pair of tokens, the dot product of the embeddings of their corresponding electrodes' ROIs will be added to the positional bias, shown in the second term of Eq. (4.3). The dot product is used instead of cosine similarity. This allows the model to assign high attention to certain regions by allowing them to have large embedding values.

The architecture of SwinTW is shown in Fig. 4.1. The input ECoG signal with a size of $T \times N$ is partitioned into $(\frac{T}{W} \times N)$ patches, each with a patch size of $W \times 1$. A linear patch embedding layer maps each patch to a C dimensional token. The SwinTW has three stages with 2, 2, and 6 layers. Swin Transformer Block (consists of a windowed multi-head self-attention layer and an MLP) is applied in each layer, detailed in [35], and we replace the spatial-temporal windowing with temporal-only windowing. Following the Swin Transformer, the temporal window partition is shifted for every two consecutive layers to allow inter-window information exchange, detailed in [35]. SwinTW performs temporal patch merging after the first and second stages, each stage decreasing the token number by half and doubling the latent dimension. After stage 3, an MLP is applied to decrease the $4C$ latent dimension to C'. Spatial max pooling across

the electrodes is then applied to convert $(\frac{T}{4W} \times N) \times C'$ feature maps to $\frac{T}{4W} \times C'$. Transposed temporal convolutions are then employed to upsample $\frac{T}{4W} \times C'$ to $T \times C'$, where T is the frame number of the input neural signal. As shown in 4.1, the $T \times C'$ latent from SwinTW next goes through the prediction head consisting of temporal convolutions (kernel-size=3) and an MLP to predict the 18 speech parameters at every frame.

In our study, we set $C = 96$ and $C' = 32$. Patch-size $W = 4$ and window size $W_t = 4$ are applied to partition the temporal dimension. In our 3 stages SwinTW with 2, 2, and 6 layers, the self-attention layers in the 3 stages have 3, 6, and 12 attention heads, respectively. The MLP in SwinTW has 3 layers (384→196→96→32) with layer norm [76] and LeakyRELU activation in between. The transposed convolution for temporal upsampling contains 4 1D transposed convolutional layers with stride=2 and kernel-size=3, padding=1. These parameter choices are determined through empirical trials and errors.

4.2.3 Training of Subject-Specific Neural Decoders

The training procedures for both the Speech Encoder and Speech Synthesizer follow the methods described in our previous work (Chapter. 3). Therefore, we omit the details in this section. Following Chapter. 3, we use two types of supervision to guide the training of the Neural Decoder that predicts speech parameters from neural signals. Firstly, we train the decoder to generate speech parameters that match the parameters generated by the speech encoder. Besides, the ground truth speech spectrograms act as additional supervision for the decoder, as the predicted speech parameters are converted to spectrograms by the speech synthesizer. The fact that our Speech Synthesizer is differentiable enables us to use the spectrogram reconstruction loss for end-to-end training.

The reference loss $L_{reference}$ for the speech parameters is defined as:

$$L_{reference} = \sum_{i,t} \lambda_i ||\hat{C}_i^t - C_i^t||_2^2,$$

i $\in [f_0^t, f_1^t, ..., f_6^t, a_1^t, ..., a_6^t, f_u^t, b_u^t, a_u^t, \alpha^t, L^t]$(4.4) where $\hat{C}_i^t$ and C_i^t are speech parameters generated by the Neural Decoder and the Speech Encoder (as ground truth), respectively. We assign each speech parameter with individual weight λ_i through testing the performances on three hybrid-density participants with different parameter choices, and the values are detailed in Chapter. 3. For spectrogram-based supervision, we use modified multi-scale spectral loss L_{MSS}, Short-Time Objective Intelligibility (STOI) loss L_{STOI}, and supervision loss $L_{supervision}$. L_{MSS} is inspired by [19]. It supervises speech reconstruction by measuring the distance between the ground truth spectrogram and the reconstructed spectrogram in both linear and mel-frequency scales. L_{STOI} measures the intelligibility of reconstructed speech based on the STOI+ metric [28]. Higher STOI+ indicates better intelligibility, the L_{STOI} is defined as the negative of STOI+: $L_{STOI} = -STOI+$. Besides, additional supervision $L_{supervision}$ is applied to improve the prediction accuracy for the pitch f_0^t and formant frequencies $f_{i=1,2,3,4}^t$. The $L_{supervision}$ calculates the L2 distance between each predicted frequency and the corresponding frequency extracted by the Praat method [45]. The overall loss for training the Neural Decoder is

$$L = L_{MSS} + \lambda_1 L_{STOI} + \lambda_2 L_{supervision} + \lambda_3 L_{reference} \tag{4.5}$$

where λ_1, λ_2, and λ_3 are set to 1.2, 0.1, and 1.0, following Chapter. 3 through testing the performances on three hybrid-density participants with different parameter choices.

Adam optimizer [77] with learning-rate=5×10^{-4}, β_1=0.9 and β_2=0.999 is used to train the Neural Decoder. As mentioned in Section 4.3.1, following Chapter. 3, randomly selected 50 out of 400 trials are used as the test set for each subject, and the remaining data are used for training.

4.2.4 Multi-Subject Neural Decoder Training

The proposed SwinTW allows the Neural Decoder to take input with any electrode layout as long as we know each electrode's MNI coordinate and region index. Therefore, this architecture enables the Neural Decoder to be trained using data from multiple participants and then used for inference on any participant. Fig. 4.1 demonstrates the multi-subject Neural Decoder training pipeline. Given participant data, a shared SwinTW-based Neural Decoder generates speech parameters based on each participant's electrode signals and electrode locations (electrodes' MNI coordinates and region index). Reference loss is calculated between the predicted speech parameters and the speech parameters generated by the subject-specific Speech Encoder. Each subject's predicted speech parameters are fed into the corresponding subject-specific speech synthesizer to create a speech spectrogram. The neural signals and electrodes' locations are fed into the Neural Decoder to generate speech parameters during inference. The participant's speech synthesizer then generates a speech spectrogram from the predicted speech parameters. Note that the embeddings for different ROIs are also learned as part of the Neural Decoder training. When we train a decoder using participants with left and right hemisphere electrodes, separate region embeddings are learned for the left and right brain hemispheres.

4.3 Results

4.3.1 Neural Data Collection and Preprocessing

The study includes 52 native English-speaking subjects (43 subjects with ECoG electrodes, 20 males, 23 females; 9 subjects with only sEEG electrodes, 3 males, 6 females) with refractory epilepsy (a disease involving seizures caused by abnormal electrical brain activity). Details about speech and ECoG signals collection can be found in Chapter. 3.

All electrodes were implanted to capture clinically relevant brain regions, detailed in Chapter. 3. There were 43 subjects who had 8x8 ECoG electrodes with 10 mm spacing capturing signals over the perisylvian cortex (male left hemisphere: 14 subjects; female left hemisphere: 13 subjects; male right hemisphere: 6 subjects; female right hemisphere: 10 subjects). Besides the 8x8 grid electrodes, some subjects had additional electrode strips outside the 8x8 grid and/or depth electrodes implanted under the brain's surface. We also included 9 subjects with only sEEG electrodes (male = 3, female = 6). This study also applies a Savitzky-Golay filter [78] with a 3rd-order polynomial and window size of 11 to further denoise the high-gamma signal in the temporal dimension. Among the 400 trials of ECoG signals recorded from the five-word production tasks, 350 trials were used for model training, and 50 trials were held out for testing (10 randomly selected trials were reserved for testing for each task).

4.3.2 Subject-Specific Models: Speech Decoding with Electrodes on One ECoG Grid

To compare our proposed grid-free SwinTW with the Neural Decoders based on ResNet and 3D Swin transformer in our previous study Chapter. 3, firstly, we

evaluated the SwinTW trained with 64 ECoG electrodes for each subject individually. Fig. 4.2 compares the decoding performance of the SwinTW decoder with the 3D ResNet and 3D Swin decoders. For each subject, we compute the average PCC, STOI+, and MCD among all the test trials. Each dot in a box plot is the mean metric for one subject. As illustrated in Fig. 4.2, the SwinTW (mean PCC = 0.825, STOI+ = 0.309, MCD = 2.341) outperforms ResNet (PCC = 0.804, STOI+ = 0.264, MCD = 2.374) and 3D Swin transformers (PCC = 0.785, STOI+ = 0.216, MCD = 2.425) in terms of PCC, STOI+, and MCD. Statistical tests using the Wilcoxon two-sided signed-rank test further show that these improvements are significant with $P<0.001$.

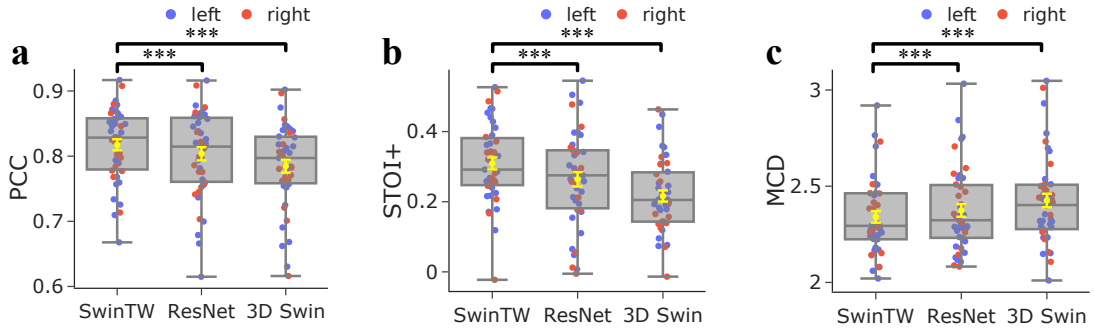


FIGURE 4.2: Subject-specific models when trained and tested on grid electrodes using different Neural Decoder architectures. Comparison of the distributions of decoding PCC **(a)**, STOI+ **(b)**, and MCD **(c)** over 43 participants. Each dot in a box plot indicates the mean metric for one participant across all testing trials. The yellow error bars denote the mean $\pm$ standard error of the mean (SEM) across participants. The SwinTW outperforms the ResNet and 3D Swin Transformer regarding PCC and STOI+. All box plots depict the median (horizontal line inside box), 25th and 75th percentiles (box), 25th or 75th percentiles $\pm 1.5\times$ interquartile range (whiskers) across all participants (N=43), and the yellow error bars denote the mean $\pm$ standard error of the mean (SEM) across participants. Distributions were compared with each other as indicated. Black brackets indicate that two experiments are compared using the Wilcoxon two-sided signed-rank test. ***: $P < 0.001$, *: $P < 0.05$, ns: $p > 0.05$.

Additionally, the performance of the three models tested on shuffled data

(by randomly shuffling the input neural signals temporally during the entire recording session) is also reported as a control in the Fig. 4.3. It is evident that the decoding performance with non-shuffled data is significantly better. Note that SwinTW differs from 3D Swin primarily in how the spatial positions of two electrodes affect the spatial attention bias between the two electrodes. With 3D Swin, the relative position between the two electrodes on the 2D grid determines the attention bias, whereas, with SwinTW, the attention bias depends on the MNI coordinates and ROI embeddings of these electrodes. Our results suggest that using the MNI coordinates and ROI information can lead to better decoding performance while making the model applicable to non-grid electrodes.

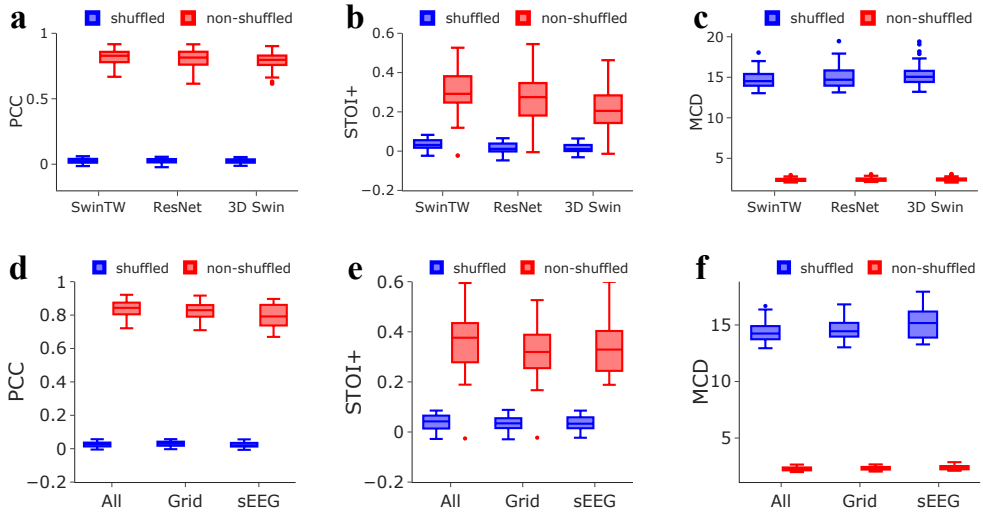


FIGURE 4.3: | To estimate chance-level performance, we temporally shuffled the input signals and fed them into a model trained with aligned data. The resulting performance metrics reflect the model's response to uninformative input. All decoding performance plots include both the original and shuffled input results for comparison.

We also assessed the decoded speech intelligibility using the state-of-the-art ASR tool Whisper [79] and report the results in Fig. 4.4. The results demonstrate that SwinTW improves speech intelligibility over ResNet and the 3D Swin

Transformer decoder.

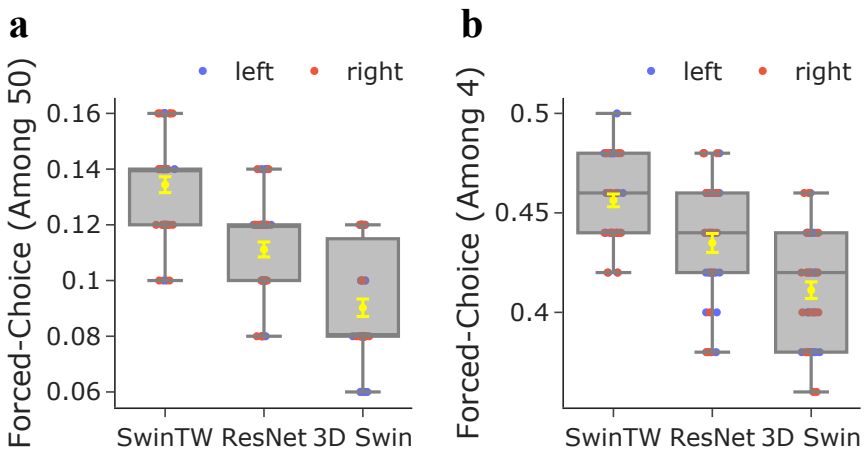


FIGURE 4.4: | **a** Forced-choice accuracy among 50 words. **b** Forced-choice accuracy among 4 words.

4.3.3 Subject-Specific Models: Speech Decoding with Additional Electrodes

As the SwinTW does not rely on the 2D grid positions of the electrodes, the proposed SwinTW can easily leverage off-grid electrodes to provide additional information for speech decoding. In our study, for each participant with additional electrodes beyond one ECoG grid, we selected additional electrodes with a standard deviation of the signal greater than a subject-specific threshold, determined following the approach described in [80] for identifying active electrodes. We then trained the SwinTW Neural Decoder with 64 electrodes from the 8x8 grid and the selected additional electrodes for each subject. As 4 participants did not have any additional electrodes that fulfill the threshold requirement, we compared the models based on the remaining 39 participants. Each participant had 1 to 19 strip electrodes, 1 to 21 depth electrodes, 1 to 21 extra grid electrodes, and 1 to 11 electrodes with unknown locations (we set the MNI coordinates of

these electrodes as 0 and the region index corresponding to Unknown instead of discarding them). Fig. 4.5 compares the SwinTW Neural Decoder performance using all selected electrodes and with the performance using only electrodes on one ECoG grid. The results demonstrate that additional electrodes can further improve the decoding performance (all electrodes: mean PCC = 0.838, STOI+ = 0.359, MCD = 2.228; grid electrodes: mean PCC = 0.825, STOI+ = 0.318, MCD = 2.341). Wilcoxon's two-sided signed-rank test further shows that these improvements are significant with P= 0.00025.

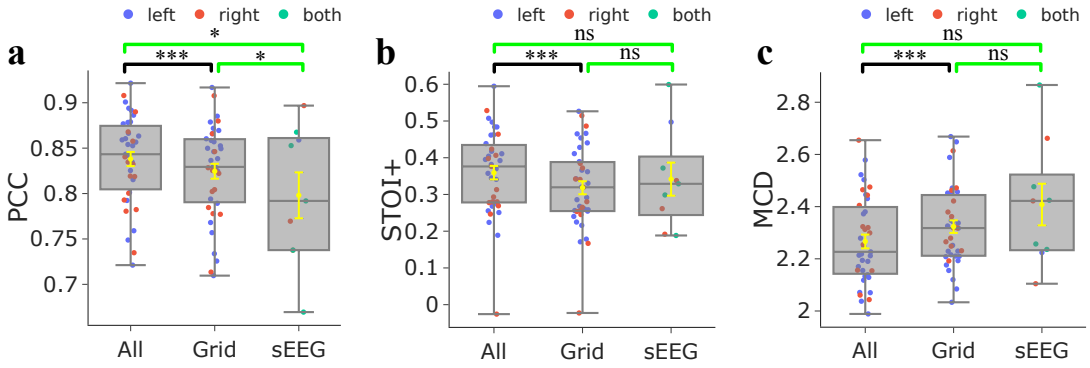


FIGURE 4.5: Decoding PCC **(a)**, STOI+ **(b)**, and MCD **(c)** shows the comparison between subject-specific SwinTW Neural Decoder performance obtained with all selected electrodes, with only electrodes on one 8x8 grid for 39 participants, and with sEEG-only electrodes over 9 participants. All electrodes outperform grid electrodes or sEEG-only electrodes. Black brackets indicate two experiments are compared using the Wilcoxon two-sided signed-rank test. Green brackets indicate two experiments that are compared using the Wilcoxon rank-sum test, indicated in green. ***: $P < 0.001$, *: $P < 0.05$, ns: $p > 0.05$.

4.3.4 Subject-Specific Models: Speech Decoding with sEEG electrodes only

We also attempted to train the proposed SwinTW model to decode speech production from only SEEG electrodes. Our study included 9 subjects (male = 3,

female = 6) with only sEEG electrodes implanted. For each subject, electrodes with a standard deviation of the signal greater than a subject-specific threshold derived following the approach of [80] were included. The number of selected electrodes for each participant ranges from 19 to 178. Fig. 4.5 demonstrates that the SwinTW can achieve promising speech production prediction based on sEEG electrodes only, with the mean of PCC slightly lower than using grid electrodes (0.798 vs 0.825), mean MCD marginally higher (2.396 vs. 2.341), but STOI+ slightly higher (0.341 vs 0.318). Note that the decoding from 64 ECoG grid electrodes is from a different patient cohort consisting of 43 participants. Wilcoxon rank-sum tests comparing these two cohorts demonstrate that the decrease in PCC is significant with $P = 0.035$. However, the differences in STOI+ and MCD are not statistically significant.

4.3.5 Multi-Subject Model: Evaluation on Test Trials of Participants within the Training Set

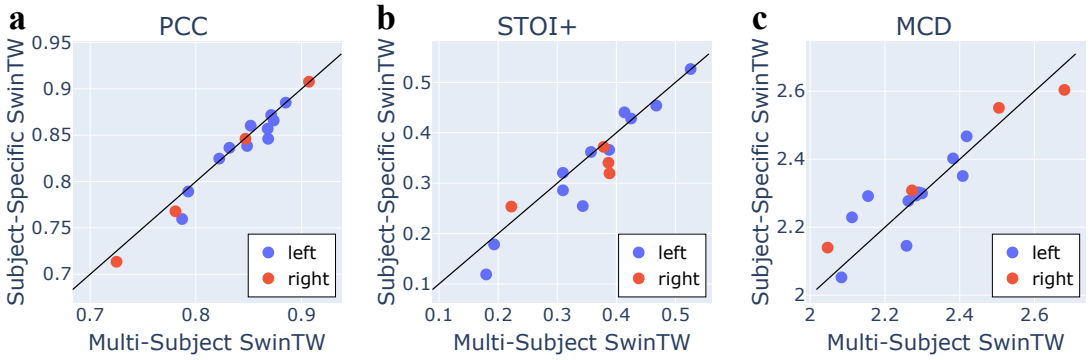


FIGURE 4.6: Comparison between a single SwinTW Neural Decoder trained with 8x8 ECoG data from training trials of multiple (15) subjects and 15 subject-specific SwinTW models. PCC, STOI+, and MCD were evaluated on test trials from the same 15 participants. Wilcoxon's two-sided signed-rank test is used to compare the two models. P-values of PCC, STOI, and MCD are 0.12, 0.06, and 0.27.

As the proposed SwinTW architecture does not require the electrodes to be arranged in a grid but relies on the electrode position in the brain, it can handle the differences in the electrode placements among different participants and allow a single model to be trained with multiple patient data. To validate this idea, we trained a single SwinTW decoder with 15 randomly selected male participants with ECoG electrodes implanted in either the left or right brain hemisphere (4 on the left and 11 on the right). As detailed in Section 4.2.4, subject-specific speech encoders and speech synthesizers are applied while the Neural Decoder is shared among subjects. We compare the decoding performance of the multi-subject and subject-specific models on the test trials of each of the 15 participants included in the multi-subject model training. As shown in Fig. 4.6, the multi-subject SwinTW model (PCC = 0.837, STOI+ = 0.352, MCD = 2.307) showed similar performance with the subject-specific model (PCC = 0.831, STOI+ = 0.334, MCD = 2.313), with no statistically significant differences.

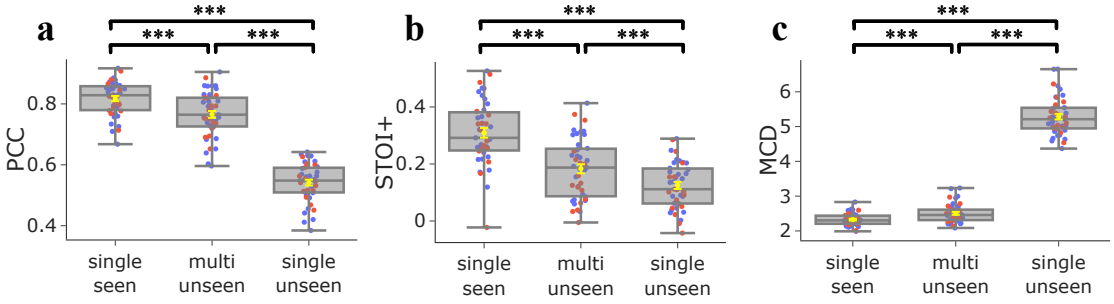


FIGURE 4.7: The decoding performance of the trained multi-subject models on ECoG participants outside the training set. Cross-validation was conducted on male and female subjects separately. **single seen** refers to subject-specific model performance on each single subject (N=43), **multi unseen** refers to the multi-subject model tested on unseen subjects, **single unseen** refers to the top five subject-specific models (based on CC) tested on unseen subjects with the same gender. Distributions were compared with each other as indicated using the Wilcoxon two-sided signed-rank test for **a**, **b**, and **c**. ***: $P < 0.001$.

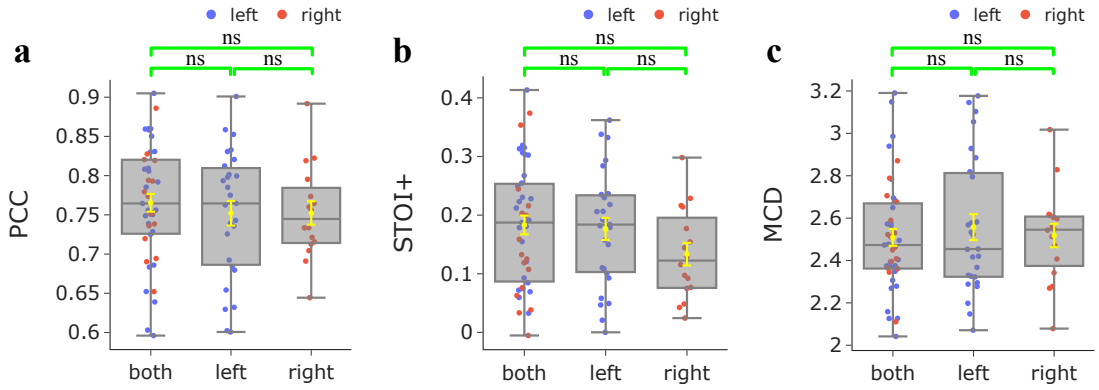


FIGURE 4.8: The comparison of speech decoding performance on unseen subjects between SwinTW trained on data from one hemisphere(left or right) and SwinTW trained on data from both hemispheres. Models were trained separately for males and females. The results demonstrate that, compared with hemisphere-specific models, the SwinTW Neural Decoder trained on both hemispheres can achieve comparable performance when inference on unseen subjects. Box plots as described in Fig. 4.2. Distributions were compared with each other as indicated. Green brackets indicate two experiments that are compared using the Wilcoxon rank-sum test. ns: $P > 0.05$.

4.3.6 Multi-Subject Model: Evaluation on Participants Outside the Training Set

We also evaluated the multi-subject SwinTW decoder on test trials of the subjects outside the training set. We conducted 5-fold cross-validation separately for male (n=20) and female (n=23) participants. Specifically, we partitioned all male (resp. female) participants (with ECoG electrodes implanted in either the left or right brain hemisphere) into five folds. Each time, we used data from four-fold participants to train a SwinTW decoder and evaluate its decoding performance on the remaining one-fold participants. The process is repeated to use every fold as the validation fold once. As shown in Fig. 4.7, although the performance achieved for participants outside the training set is lower than the subject-specific models, the decoded speech still has a high mean PCC of 0.765.

We also tested the top-5 single-subject model (with top-5 performance based on CC on subject-specific experiments) on unseen subjects. More specifically, we selected three top-performing subject-specific models for female participants and evaluated each of them on unseen female (n=22) subjects. Similarly, we selected two top-performing subject-specific models for male participants and tested each on unseen male (n=19) subjects. The results indicate that subject-specific models perform significantly worse than multi-subject models on unseen participants. While the performance of **multi unseen** is lower than that of **single seen** (subject-specific models on the testing trials of the same participants used for training), it is still significantly better than the **single unseen** model (subject-specific models tested on other participants). These results demonstrate that the proposed SwinTW decoder trained with multiple subject data can achieve generalizability for participants who are unseen during model training, while subject-specific models cannot.

To investigate if separate models should be trained for decoding from neural data in the left and right hemispheres, we performed additional experiments, where we trained and evaluated multi-subject models for the two hemispheres separately, each through cross-validation. Among male participants, there were 14 with left hemisphere data and 6 with right hemisphere data. For female participants, we had 13 with left hemisphere data and 10 with right hemisphere data. We used 5-fold cross-validation for training and evaluating each model. As shown in Fig. 4.8, compared with hemisphere-specific models, the SwinTW decoder trained using data from both hemispheres achieved comparable performance when tested on unseen subjects. This suggests that a single SwinTW model can effectively extract and synthesize information from both hemispheres for speech decoding. We have also added the audio demos containing sample decoded speech in the `https://xc1490.github.io/swinTW`.

4.4 Conclusion and Discussions

This study proposes a new Neural Decoder architecture, SwinTW, that does not have the grid-input assumption and can predict speech parameters from electrodes in any topological layout in the brain. The SwinTW removes the grid-based operations in the 3D Swin Transformer model used in our prior study (Chapter. 3) to make the model applicable for electrodes in any layout. Instead of relying on 2D grid indices to provide positional information about each electrode, the SwinTW relies on each electrode's position in the standardized brain coordinate (i.e., MNI) and the brain region that the electrode resides in to generate relative positional bias for self-attention. The SwinTW was used as the Neural Decoder in the speech decoding pipeline proposed in our previous work and was trained using the 2-step training pipeline in Chapter. 3. Our proposed SwinTW Neural Decoder achieved superior performance over the Neural Decoders based on ResNet and 3D Swin Transformer in Chapter. 3, which can only work with ECoG data. As illustrated in Fig. 4.2, over 43 participants with low-density 8x8 ECoG electrodes, the SwinTW achieved higher mean PCC, STOI+, and lower MCD (PCC: 0.825, STOI+: 0.309, NCD: 2.341) than the ResNet (PCC:0.804, STOI+: 0.264, MCD: 2.374) and 3D Swin Transformer (PCC: 0.785, STOI+: 0.216, MCD: 2.425) using the same 64 electrodes from the 8x8 ECoG grid. We attribute SwinTW's better performance to its utilization of electrode' locations on the brain cortex (the MNI coordinate and brain region information) rather than the 2D grid index.

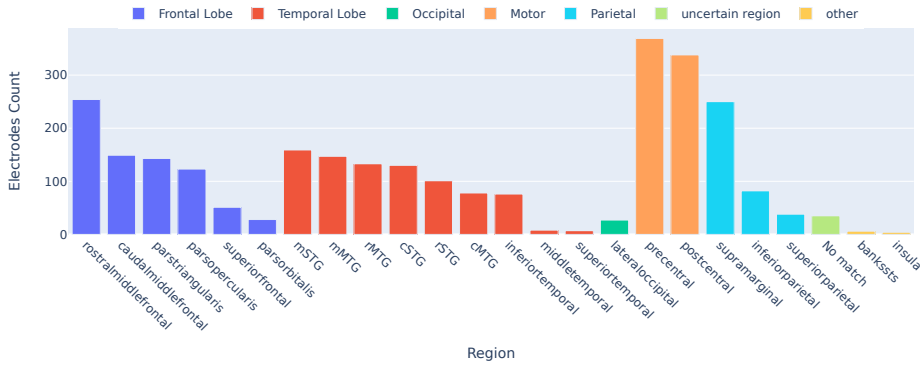
Unlike ResNet and 3D Swin Transformer, the SwinTW does not rely on 2D grid indices of electrodes and can accommodate both ECoG electrodes, strip and depth electrodes, and additional grid electrodes. Our results demonstrate

that leveraging the additional electrodes can improve speech decoding performance, as illustrated in Fig. 4.5. Specifically, for 39 subjects with additional active electrodes, the SwinTW utilizing the additional electrodes achieved better mean PCC (0.838), STOI+ (0.359), and MCD (2.228) compared with the SwinTW using grid electrodes only (PCC: 0.825, STOI+: 0.318, MCD: 2.341). The superior results indicate that the neural activity recorded by the additional electrodes contains complementary information for decoding speech.

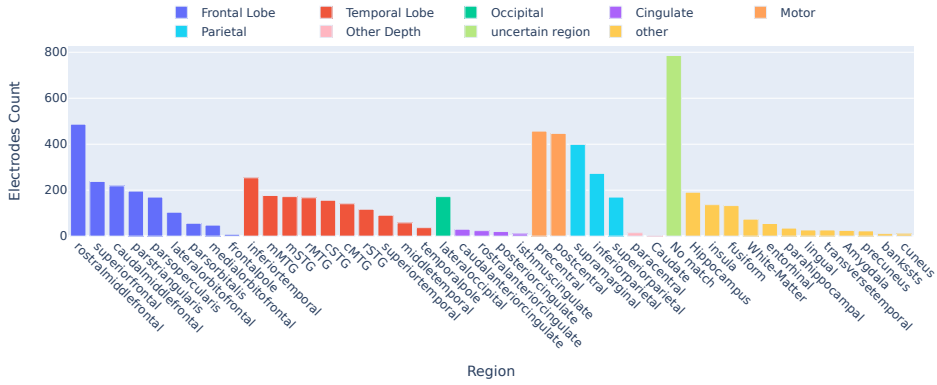
Our results further demonstrate that the SwinTW can achieve high decoding quality based only on sEEG electrodes. Specifically, as shown in Fig. 4.5, for nine subjects with only sEEG electrodes implanted, we achieved mean PCC: 0.798, STOI+: 0.341, and MCD: 2.396. The mean and range of PCC are slightly lower than the decoding performance obtained using ECoG electrodes but significantly higher than previously reported decoding performance from sEEG only, ranging between 0.54 to 0.77 in mean PCC [24, 55, 81, 82]. It is notable that there is no statistical difference between the decoding performance from sEEG vs. from ECoG or all electrodes in terms of STOI+ and MCD, the metrics that are better indicators of the intelligibility of the decoded speech.

We present the distribution of electrode coverage in Fig. 4.9, alongside the electrode contribution analysis in Fig. 4.10, which quantifies the importance of each electrode across all participants. The methodology employed for the contribution analysis is detailed in Chapter. 3. The contribution analysis is conducted using subject-specific models. Despite the variation in electrode coverage across the grid, all-electrode, and sEEG cases, the contribution analysis reveals consistent patterns. Specifically, electrodes in the motor and temporal lobe regions exhibit greater contributions compared to those in other regions. Conversely, electrodes in other and unidentified regions show the lowest contributions, despite the relatively large number of electrodes present in these areas. This may

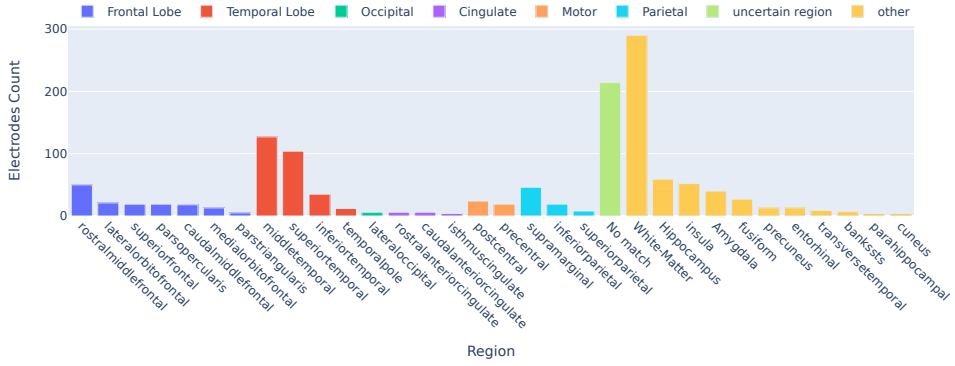
explain why similar decoding performance can be achieved despite variations in electrode coverage. That is, the sEEG is sampling sufficient cortical regions as ECoG, thus leading to similar decoding performance.



(A) Grid electrodes distribution over the brain across 43 Participants with ECoG electrodes

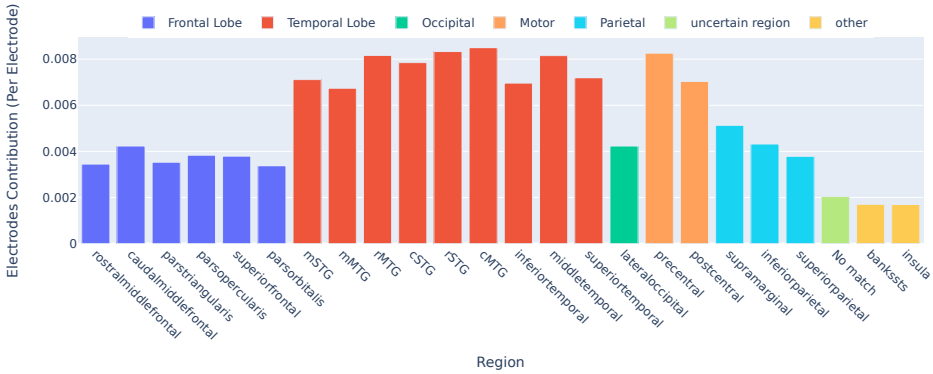


(B) All electrodes distribution over the brain across 43 Participants with ECoG and other electrodes

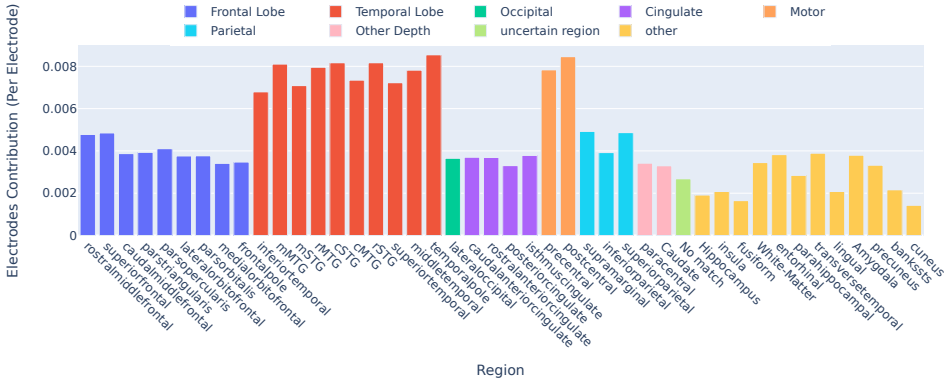


(C) Depth electrodes distribution over the brain across 9 Participants with sEEG electrodes only

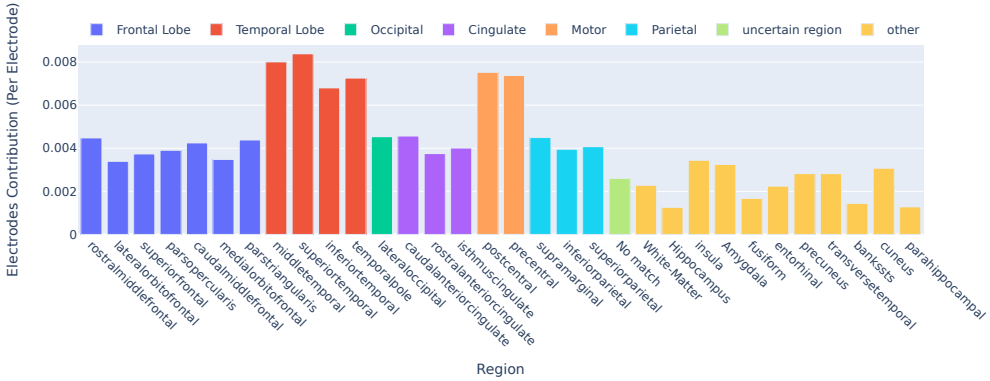
FIGURE 4.9: Electrodes Distribution



(A) Grid electrodes contribution averaged over 43 Participants



(B) All electrodes contribution averaged over 39 Participants



(C) Depth electrodes contribution averaged over 9 sEEG Participants

FIGURE 4.10: Electrodes Contribution

Since the SwinTW directly uses the anatomical positions of electrodes rather than their grid indices, it can be trained with data from multiple subjects. As shown in Fig. 4.6, when evaluated on the testing trials of the participants in the training cohort and using only data on 8x8 grids, the resulting multi-subject

model trained with data from 15 participants achieved statistically on par decoding performance (mean PCC: 0.837, STOI+:0.352, MCD: 2.307), compared with the SwinTW trained for each subject individually (mean PCC: 0.831, STOI+: 0.335, and MCD: 2.313). This implies that the SwinTW model structure is able to effectively deal with the significant variability in the electrode placements among patients and make use of electrodes' positions on the cortex. Previously, we have attempted to train ResNet and 3D Swin-based Neural Decoders using ECoG data from multiple participants. We obtained significantly worse decoding performance compared to subject-specific models. That is likely because ResNet and 3D Swin models rely on the electrodes' relative positions in the 2D grid. Because the ECoG grid is placed differently among the participants, the same relative difference in the 2D grid can be associated with very different anatomical positions in different participants, making using the grid index as positional information unsuitable when the data come from multiple participants.

Most significantly, the SwinTW model trained with multiple participants' data demonstrated generalizability to participants outside the training cohorts, with a high average decoding PCC (mean PCC = 0.765 over 43 unseen participants through a cross-validation study conducted separately for males and females). Fig. 4.7 shows that the speech decoding performance achieved on unseen subjects overlaps significantly with that of the subject-specific model. Testing on unseen subjects presents a significantly more challenging task than evaluating the model on subjects whose data were used during training. Despite the increased difficulty, our SwinTW model still leads to decoding evaluation metrics that are significantly above chance (shown in Fig. 4.10). We consider this

reasonable performance as a demonstration of the generalizability of our approach, even though the performance on unseen participants is lower than using subject-specific models. Furthermore, a model trained with data from both the left and right hemispheres performs on par with those trained using only the left or right hemisphere on unseen participants (Fig. 4.8). These results suggest that the SwinTW training using multiple participants' data can successfully learn how to handle differences among subjects based on electrode signals and the anatomical position of the electrodes. The success of the left and right hemispheres co-training demonstrates the strong learning capacity of the SwinTW. The two-hemisphere co-training also allows the Neural Decoder to fully leverage the whole dataset, as we no longer need to train the model separately for each hemisphere.

To summarize, the SwinTW Neural Decoder can predict speech parameters from electrode signals and electrodes' positions on the brain cortex without requiring the electrodes to be arranged in a grid. The SwinTW Neural Decoder, in conjunction with our previously reported Speech Synthesizer, demonstrated superior speech decoding performance compared with our prior works based on ResNet and 3D Swin Transformers when only electrodes on a single ECoG array were used. Besides, the grid-free architecture of the SwinTW allows the model to leverage off-grid electrodes to improve speech decoding further. When using only sEEG data, the decoding performance was comparable to that using ECoG data. As explained in the Introduction, decoding speech from sEEG signals would have significant clinical advantages over using ECoG data. Furthermore, the SwinTW can be trained with data from multiple subjects, regardless of whether the electrodes were implanted in the left or right brain hemispheres. The multi-subject SwinTW performed statistically on par with the

subject-specific models for participants within the training cohort. Most importantly, our SwinTW trained with multiple participants' data demonstrated good generalizability to subjects outside the training cohorts, achieving high average decoding PCC.

We are one of the few studies ([9, 10, 75]) demonstrating speech-decoding models trained across multiple participants. However, these other prior works embed subject-specific layers in their model structures, and hence, the models need subject-specific data for training. To our knowledge, we are the first to design a framework that goes beyond subject-specific training without using subject-specific layers. Our result demonstrates the exciting possibility of developing speech prostheses without collecting subject-specific training data: We can train a reliable decoder with data from selected participants and then directly deploy the model to a new participant. Note that although our experiments on the multi-subject model only considered the ECoG grid data, we expect similar trends when using ECoG plus non-grid data or non-grid data only.

Notably, the proposed SwinTW Neural Decoder is not limited to using our speech synthesizer. It could potentially be used to decode other latent features, e.g. the HuBERT latent features [17], which can then drive a corresponding synthesizer [83]. The work in [14] successfully decoded speech with high word decoding accuracy by decoding to quantized HuBERT units using an RNN decoder from high-density ECoG signals of a single participant. However, the RNN structure cannot be trained with multi-subject data without introducing subject-specific layers. It will be interesting to explore the potential of training a SwinTW decoder using data from multiple participants with surface and/or depth electrodes, to map the neural signals to the HuBERT units and compare the decoding performance with the subject-specific RNN model or multi-subject RNN models with subject-specific layers.

One limitation of our current study is that the decoding performance for participants outside of the training cohorts is not consistently high. This could be potentially solved by including more participants in the training set when larger datasets become available. Furthermore, the SwinTW model structure can also be extended to include subject-specific layers for improved performance. We would explore training the non-subject-specific layers with a large pre-collected multi-subject dataset and refining only the subject-specific layer with a small amount of data for any new participants.

The work is in close collaboration with Dr. Junbo Chen. I was responsible for the model development, evaluation, and visualization. During the project, Junbo also contributed extensively to model development, training, and evaluation, and together we wrote a paper based on the results of this chapter, which has been published in the Journal of Neural Engineering [21].

Chapter 5

Decoding Speech from sEEG Recordings via HuBERT- and Articulatory-Representation–Based Synthesizers

5.1 Introduction

Recently, significant strides have been made in developing high-performance and intelligible speech prostheses [14] by leveraging high-density ECoG and Utah array recordings with the HuBERT-based speech synthesizer. However, while these approaches benefit from high spatial resolution, their invasiveness limits clinical and research applicability. By contrast, stereoelectroencephalography (sEEG) provides a slightly less invasive recording modality that does not suffer from complications of a large craniotomy. This approach has also become increasingly accessible in the clinical setting. Previous work has demonstrated real-time synthesis of imagined speech from sEEG [15], although the resulting speech has remained unintelligible due to simplified decoding pipelines and

limited exploitation of the temporal nature of neural signals.

In light of these advances and challenges, this chapter investigates how other latent representations can improve decoding speech from sEEG signals. We focus on two complementary approaches: HuBERT and articulatory space.

First, we reformulate neural speech decoding as a classification task using HuBERT [17], a self-supervised speech representation model trained to predict cluster assignments by doing mask encoding on speech representations. By decoding neural signals into HuBERT token sequences, we harness pretrained linguistic priors to enhance intelligibility, particularly when working with limited and noisy sEEG datasets. The Hubert encoder, which integrates a convolutional front-end and a BERT-based transformer, produces context-aware representations that have shown strong performance in automatic speech recognition tasks and can be adapted for neural decoding by attaching a Tacotron2 [84] and WaveGlow-based [85] speech vocoder.

Second, we leverage the articulatory space [18], a continuous and physiologically grounded latent space defined by the spatiotemporal dynamics of six primary articulators and two source features (pitch and loudness), for speech decoding. This low-dimensional, interpretable latent representation offers several advantages: it directly captures the physical mechanisms of speech production, retains speaker acoustic characteristics through the use of a speaker embedding, and is robust to noise and cross-speaker variability. Articulatory space-based speech synthesis leverages the pretrained WavLM network to extract features from original speech, and employs a linear mapping from the features to articulatory trajectories, followed by HiFi-GAN synthesis.

Compared to the source-filter speech synthesizer developed in Chapter 3, the HuBERT and articulatory space approaches offer complementary advantages.

The source-filter model explicitly encodes spectral properties based on classical speech production theory, HuBERT provides data-driven linguistic representations learned through self-supervision on large speech corpus, and the articulatory space captures speaker-specific kinematic dynamics grounded in vocal tract physiology. Together, these approaches offer alternative and enhanced pathways for speech decoding, addressing key trade-offs among interpretability, generalization, data efficiency, and clinical applicability.

In this chapter, we systematically evaluate these representations for neural speech decoding, comparing their performance on sEEG data and examining their potential to advance the development of scalable, interpretable, and realistic speech neuroprostheses.

5.2 Methods

5.2.1 HuBERT representation of speech

The HuBERT approach uses multiple iterations of k-means clustering to predict the hidden cluster assignments of the masked frames ($o_3^{'}, o_4^{'}, o_5^{'}$ in the Fig. 5.1). The mappings alternate iteratively between step 1 and step 2. In step 1, the clustering feature extraction module processes the audio to form the MFCC feature and clusters each frame to generate its hidden embedding by k-means clustering. In step 2, the audio is fed to a CNN encoder to generate latent representations for each frame. Some consecutive frames are masked and all frames are fed to a transformer. The transformer tries to learn the contextual information and recover the masked features o_3, o_4, o_5. They are further projected to $o_3^{'}, o_4^{'}, o_5^{'}$. The hidden unit embeddings compute cosine similarity with quantized units as logits. The cross-entropy loss between the logits and the quantized units is used

to train the CNN encoder, transformer, and project layers. After the first iteration, the feature embeddings of the sixth layer are used as features for clustering instead of MFCC for better clustering.

The HuBERT encoder follows the wav2vec 2.0 architecture [86], with a convolutional waveform encoder, a BERT encoder [87], a projection layer and a code embedding layer. The waveform encoder is composed of seven 512-channel layers with strides [5,2,2,2,2,2,2] and kernel widths [10,3,3,3,3,2,2]. The BERT encoder consists of many identical transformer blocks. The convolutional waveform encoder generates a feature sequence at a 20ms frame interval for audio sampled at 16kHz (CNN encoder down-sampling factor is 320x). The audio-encoded features are then randomly masked. The BERT encoder takes as input the masked sequence and outputs a feature sequence $[o_1^{'}, \cdots, o_T^{'}]$. The distribution over codewords is parameterized with

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\mathrm{sim}(A^{(k)} o_t^{'}, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(A^{(k)} o_t^{'}, e_{c'})/\tau)}, \tag{5.1}$$

where A is the projection matrix, e_c is the embedding for codeword c, $\mathrm{sim}(\cdot,\cdot)$ computes the cosine similarity between two vectors, and τ scales the logit, which is set to 0.1. When cluster ensembles are used, one projection matrix $A^{(k)}$ is applied for each clustering model k. The distribution is used as logits and compared with the quantized units with cross-entropy loss.

After HuBERT pre-training, the HuBERT encoder could be used for downstream tasks such as Automatic Speech Recognition (ASR). Connectionist temporal classification (CTC) [88] loss could be used for ASR fine-tuning the whole model weights except for the convolutional audio encoder, which remains frozen. We use CTC loss/aligned cross-entropy/aligned focal loss to guide the sEEG Decoder to generate decoded HuBERT units to decode speech. It is important to

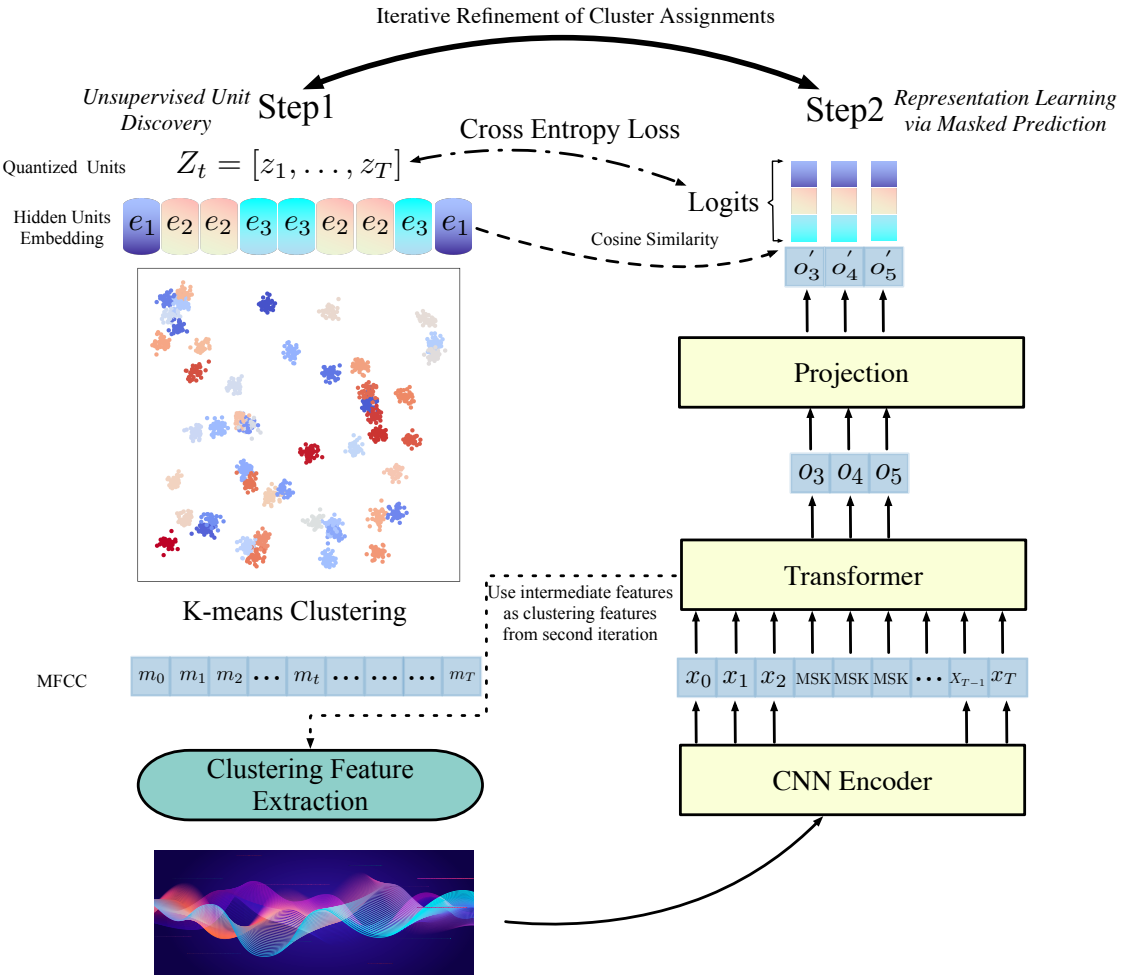


FIGURE 5.1: | **Pretraining of HuBERT using Self-Supervised Learning with feature clustering.** The HuBERT approach uses multiple iterations of k-means clustering to predict the hidden cluster assignments of the masked frames (o_3', o_4', o_5' in the figure). The assignments alternate iteratively between Step 1 and Step 2. During step 1, the clustering feature extraction module processes the audio to form the MFCC feature and clusters each frame to generate its hidden embedding by k-means clustering. In step 2, the audio is fed to a CNN encoder to generate latent representations for each frame. Some consecutive frames are masked and all frames are fed to a transformer. The transformer tries to learn the contextual information and recover the masked features o_3, o_4, o_5. They are further projected to o_3', o_4', o_5'. The hidden unit embeddings are used to compute cosine similarity with quantized units as logits. The cross-entropy loss between the logits and the quantized units is used to train the CNN encoder, transformer, and project layers. After the first iteration, the feature embeddings of the sixth layer are used as features for clustering instead of MFCC for better clustering.

note that the HuBERT units do not carry speaker-specific acoustic information. The pretrained tacotron2 and waveglow model generate speech with a particular female voice that the synthesis model was trained on.

5.2.2 Articulatory space representation of speech

Besides HuBERT representations, we sought a feature space that is more physically plausible, inherently interpretable, and more compact, yet still capable of driving realistic speech synthesis. To this end, we adopt the *articulatory space* [18], which directly encodes the spatiotemporal trajectories of six primary midsagittal articulators—upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (TT), tongue blade (TB), and tongue dorsum (TD)—each in two dimensions (X–Y), yielding a 12-dimensional kinematic signal sampled at 50 Hz. By incorporating two additional features—frame-wise fundamental frequency and loudness—the model yields a 14-channel representation that provides a compact and physiologically meaningful interface with the underlying mechanisms of speech production.

This articulatory encoding is more compact than our previous source-filter speech synthesizer in Chapter 3). Compared with HuBERT synthesizer, it also has a simple, speaker-agnostic inversion: a single linear layer maps 768-D WavLM [89] embeddings to 12-D articulator trajectories, preserving pretrained generalization while introducing minimal trainable parameters. In contrast to generic HuBERT synthesis, conditioning a neural synthesizer on articulatory features plus a lightweight speaker embedding enables explicit speaker control and yields more natural, high-fidelity speech.

Articulatory and source feature encoding

The articulatory space is defined by six key midsagittal articulators, each tracked in two dimensions (X–Y), yielding a 12-dimensional kinematic representation, together with two source features (pitch and loudness) at 50 Hz:

Upper Lip (UL) Horizontal and vertical displacement of the upper lip.

Lower Lip (LL) Horizontal and vertical displacement of the lower lip.

Lower Incisor (LI) Position of the lower front tooth, reflecting jaw opening and lip contact.

Tongue Tip (TT) Coordinates of the tongue apex, crucial for alveolar and dental consonants.

Tongue Blade (TB) Location of the tongue just behind the tip, shaping many fricatives and stops.

Tongue Dorsum (TD) Mid-tongue body position, governing vowel quality and velar articulations.

Pitch Frame-wise fundamental frequency f_0, extracted via CREPE [90].

Loudness Frame-wise RMS magnitude over 20 ms windows.

Summary of the Articulatory Space Training Framework

SPARC [18] employs a two-stage, end-to-end differentiable pipeline to resynthesize speech via interpretable articulatory parameters:

1. Acoustic-to-Articulatory Inversion (AAI)

1. **Feature Extraction:** Input audio (16 kHz) is fed through a frozen WavLM-Large encoder [89]; the 9th transformer layer's frame-wise embeddings ($\mathbf{h}_t \in \mathbb{R}^{768}$) are collected.

2. **Linear Mapping:** After down-sampling the target Electromagnetic Articulography (EMA) signal from 200 Hz to 50 Hz, utterance-level z-score normalization, and 10 Hz low-pass Butterworth filtering, a single linear layer $\mathbf{y}_t = W\,\mathbf{h}_t + \mathbf{b}, \quad W \in \mathbb{R}^{12\times 768},\ \mathbf{b} \in \mathbb{R}^{12}$, is trained using mean squared error (MSE) loss on the MNGU0 single-speaker EMA dataset to learn the mapping from WavLM features to articulatory trajectories. Only W and $\mathbf{b}$ are updated; all upstream weights remain frozen.

2. Conditional Neural Synthesis (HiFi-GAN)

1. **Condition Inputs:** $\{$12-D articulatory traces, f_0, loudness$\}$ at 50 Hz, concatenated with a 64-D speaker embedding produced by a frozen WavLM CNN + weighted pooling + small FFN.

2. **Generator Architecture:** A HiFi-GAN generator upsamples the 78-D condition vector to 16 kHz waveform via transposed convolutions and Multi-Receptive-Field Fusion blocks.

3. **Discriminators:** Multi-Scale and Multi-Period discriminators judge real vs. synthetic audio, enforcing both long- and short-term consistency.

4. **Losses:** The model is trained with a combination of:

    - Hinge adversarial loss,
    - Multi-resolution STFT reconstruction loss (L1 on magnitude and phase at multiple FFT settings),

- Feature-matching loss on discriminator activations.

This two-stage modular approach ensures that articulatory inversion generalizes across speakers via a single-speaker EMA template, while the conditioned HiFi-GAN synthesizer, trained on diverse voices, can reconstruct any speaker's waveform given their articulatory and speaker codes.

5.2.3 Source-Filter Speech Synthesizer vs. Articulatory-Space Framework

Here we want to discuss and compare the differences and similarities between our proposed Source-Filter Speech Synthesizer [20] (described in Chapter. 3) and the articulatory space framework [18].

Source-Filter Speech Synthesizer Our earlier Source-Filter Speech Synthesizer [20] uses a fully differentiable source–filter model:

- **Voiced:** Harmonic excitation $H^t(f)$ from f_0^t (over K harmonics) passed through six time-varying formant filters $F_i^t(f)$.
- **Unvoiced:** White noise through a broadband filter $F_{\hat{u}}^t(f)$ plus the same formants.

Voiced and unvoiced outputs mix via weight α_t, then scale by loudness L^t and add noise $B(f)$:

$$\widehat{S}^t(f) = L^t\left[\alpha_t V^t(f) + (1-\alpha_t)U^t(f)\right] + B(f)\,.$$

This yields 18 parameters per frame at 125 Hz: $\{f_0^t, \alpha_t, L^t\}, \{f_i^t, a_i^t\}_{i=1..6}, \{f_{\hat{u}}^t, b_{\hat{u}}^t, a_{\hat{u}}^t\}$.

Articulatory-Space Framework SPARC [18] represents speech with 14 channels at 50 Hz:

1. **Articulatory kinematics (12):** UL, LL, LI, TT, TB, TD (X/Y), obtained by a linear head mapping 768-D WavLM-Large features to EMA space.
2. **Source features (2):** Frame-wise pitch and loudness.

A HiFi-GAN vocoder conditioned on these, plus a 64-D speaker embedding, produces the waveform.

Similarities

- Both are fully differentiable.
- Both yield compact, interpretable controls (18 spectral vs. 14 articulatory/source parameters).

Differences and Advantages of Articulatory Space

- **Rich EMA supervision:** The real single-speaker EMA data provides strong and ecologically valid supervision. The MNGU0 dataset offers 75 minutes of training data for learning the linear mapping from WavLM features to articulatory trajectories, whereas the source-filter speech synthesizer only has 5 minutes of speech data to train the speech encoder.
- **Pretrained WavLM features:** Compared to the speech encoder in source-filter speech synthesizers, which is typically trained from scratch on only a few minutes of speech data, the articulatory space model leverages WavLM features pretrained on tens of thousands of hours of audio. This allows the

system to benefit from rich contextual, speaker, and noise-robust representations, while only requiring the training of a lightweight linear mapping layer to predict articulatory trajectories.

- **Speaker control and disentanglement:** The articulatory space synthesizer integrates an independent speaker embedding module, enabling explicit modeling and disentanglement of speaker characteristics. This design allows training on large-scale, multi-speaker corpora and facilitates robust speaker adaptation. In contrast, the source-filter speech synthesizer implicitly captures speaker characteristics within the prototypes of the voiced and unvoiced filters, without explicit disentanglement, and is typically not trained on cross-subject data.

- **HiFi-GAN synthesis:** Adversarial, multi-resolution STFT, and feature-matching losses yield detailed, artifact-free audio. Compared to our rule-based source-filter synthesizer, HiFi-GAN serves as a substantially more expressive vocoder with greater modeling capacity; whereas the source-filter speech synthesizer explicitly models synthesis mechanisms, HiFi-GAN leverages large-scale data to implicitly learn source-filter synthesis strategies.

Comparison between three speech representations Here we give a complete comparison between the three representations in Table 5.1. The source-filter speech synthesizer employs a continuous 18-dimensional representation explicitly derived from the source-filter model, making it highly interpretable and speaker-specific. In contrast, HuBERT uses a discrete set of 100 classes learned via self-supervised masked prediction, offering generic semantic or phonetic

representations not constrained by source-filter assumptions, but with moderate interpretability. Finally, the articulatory space relies on a continuous 14-dimensional representation capturing speaker-specific vocal tract dynamics, which is implicitly modeled through HiFi-GAN learning using audio and articulatory data. Compared to the other two, the articulatory space combines high interpretability with strong physiological grounding, making it an appealing choice for applications requiring fine-grained control over speech articulation. Overall, these three approaches represent complementary strategies in the trade-off between interpretability, data requirements, and generalization, and our work systematically explores their strengths and limitations across multiple decoding scenarios.

Model	**Source-Filter Speech Synthesizer**	**HuBERT**	**Articulatory Space**
Representation Type	Continuous	Discrete	Continuous
Dimension	18 dimensions	100 classes	14 dimensions
Voice Characteristics	Speaker-specific	Generic	Speaker-specific
Source-Filter Model Assumption	Explicitly built based on source-filter theory	No assumption	Implicitly modeled via HiFi-GAN
Training Data	Audio only	Audio only	Audio + articulatory data
Interpretability	High	Low	High

TABLE 5.1: Comparison of Speech Representation Models

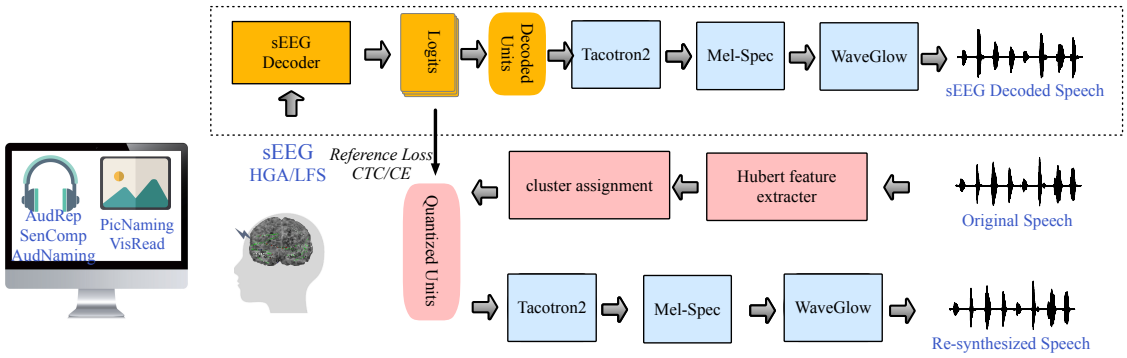


FIGURE 5.2: | **Neural Speech Decoding Framework with HuBERT Synthesizer** The proposed neural speech decoding framework. The upper part shows the sEEG-to-speech decoding pipeline. The sEEG decoder generates time-varying logits from sEEG signals, which are converted into units. A pre-trained Tacontron2 model takes the decoded units and generates a mel spectrogram, which is used by Waveglow to decode the speech. The lower part shows the pre-trained HuBERT speech synthesizer taking speech as input and generating features that are used for clustering to generate quantized units per frame. The pre-trained Tacontron2 model takes the quantized units and generates a mel spectrogram, which is used by Waveglow to resynthesize the speech. The lower speech re-synthesis part is used as a guide to generate quantized units to help the sEEG decoder map to the units as closely as possible. The sEEG decoder is a classifier that generates time-varying class labels for each frame. During inference, only the upper part is used.

5.2.4 Proposed sEEG speech decoding framework

Similar to Section 3.2, we solve the sEEG to speech decoding problems with two steps.

HuBERT guided speech decoding

Our sEEG-to-Speech pipeline in Fig. 5.2 comprises two components: an sEEG decoder that maps intracranial signals to frame-wise logits (quantized units), and a pretrained HuBERT synthesizer [17], implemented via Tacotron2 [91] and WaveGlow [85], which converts those units into mel-spectrograms and ultimately into waveforms.

HuBERT Autoencoder Pretraining We first freeze the HuBERT encoder to extract discrete units from raw speech. Pretrained Tacotron2 and WaveGlow models are then used to reconstruct the original waveform from these units (lower panel, Fig. 5.2), forming a self-supervised autoencoder.

sEEG Decoder Supervision Next, the sEEG decoder is trained to predict HuBERT's quantized units from neural signals using either a CTC loss [88] or an aligned cross-entropy/Focal loss. No additional auxiliary losses are imposed since accurate unit prediction alone is assumed sufficient to yield intelligible decoded speech.

Articulatory Space Guided Decoding

Our articulatory-space guided decoding framework (Fig. 5.3) adopts a similar setup to the HuBERT-based framework, but replaces discrete units with a 14-dimensional latent articulatory representation and employs a conditioned HiFi-GAN for waveform reconstruction.

Articulatory Analysis–Synthesis Pretraining Original speech is passed through three frozen extractors: a speaker encoder, an articulatory feature extractor (linear mapping from WavLM-Large features to EMA-derived kinematics), and a source feature extractor (CREPE for pitch, RMS for loudness). These 14-D features condition a HiFi-GAN vocoder to re-synthesize speech, and the reconstruction loss (MSE or Huber) provides a high-fidelity reference for training the sEEG decoder (lower panel).

sEEG Decoder Supervision Intracranial sEEG signals recorded during auditory/visual naming tasks are mapped by the sEEG decoder into time-varying 14-D latents. Without any auxiliary objectives, we supervise this mapping with MSE/Huber loss against the reference articulatory and source features. At inference, the decoder outputs latents and feeds them into the pretrained HiFi-GAN together with the speaker embedding(pre-extracted through the speaker's original speech) to produce intelligible, speaker-specific speech (upper panel).

5.2.5 Loss functions for HuBERT and Articulatory Representations

For our two distinct latent representations, we adopt distinct loss formulations:

HuBERT representation The HuBERT latents are quantized into $C = 100$ discrete classes, so we treat prediction as a classification task:

- In misaligned settings one typically uses Connectionist Temporal Classification (CTC) loss [88] to handle unknown alignments:

$$\mathcal{L}_{\text{CTC}}(\mathbf{X}, \mathbf{y}) = -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{y})} \prod_{t=1}^{T} p(\pi_t \mid X_t).$$

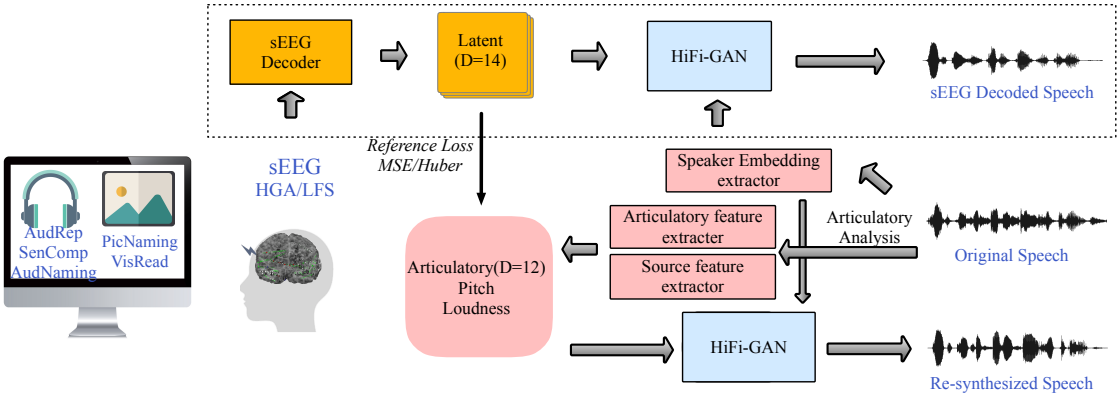


FIGURE 5.3: | **Neural Speech Decoding Framework with Articulatory Space Upper panel:** sEEG-to-speech decoding. Intracranial sEEG signals recorded during auditory/visual tasks are mapped by an sEEG decoder into time-varying latent vectors (12-D articulatory trajectories + pitch + loudness), which—together with a learned speaker embedding—condition a pretrained HiFi-GAN vocoder to produce decoded speech. **Lower panel:** Articulatory analysis–synthesis for training. Original speech is processed by separate speaker identity, articulatory feature, and source feature extractors to generate reference articulatory kinematics and source signals. These reference features drive the same HiFi-GAN to re-synthesize speech, and the reconstructed waveform provides supervisory feedback (via MSE/Huber loss) to align the sEEG-decoded latents. During inference only the upper pathway is used.

- Because our sEEG–speech pairs are precisely aligned, we could also apply standard cross-entropy loss for unit classification:

$$\mathcal{L}_{\text{CE}}(\mathbf{p}, \mathbf{y}) = -\sum_{i=1}^{C} y_i \log p_i.$$

- To address the uneven frequency of different HuBERT units, we also experiment with focal loss [92], which down-weights easy examples and focuses training on rarer classes:

$$\mathcal{L}_{\text{FL}}(\mathbf{p}, \mathbf{y}) = -\sum_{i=1}^{C} \alpha_i \, (1 - p_i)^{\gamma} \, y_i \log p_i.$$

Articulatory representation The articulatory features lie in a continuous $D = 14$-dimensional space and are therefore regressed directly:

- Under perfect alignment, we employ mean squared error (MSE) loss to minimize squared deviations.

$$\mathcal{L}_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \|\hat{\mathbf{y}}_n - \mathbf{y}_n\|^2 \tag{5.2}$$

- We additionally evaluate Huber loss [93] for its robustness to occasional large errors.

$$\mathcal{L}_{\text{Huber}}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \begin{cases} \frac{1}{2}(\hat{y}_n - y_n)^2, & |\hat{y}_n - y_n| \le \delta, \\ \delta \, |\hat{y}_n - y_n| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{5.3}$$

5.2.6 sEEG Decoder Architectures

Fourier Spatial Attention (3-D)

In addition to the channel-wise attention from Chapter. 4, we adopt the *Fourier Spatial Attention (FSA)* [75] to remap C sEEG channels into a fixed $D_1 = 270$ dimensional spatial feature space. FSA defines per-output-channel Fourier-parameterized

attention functions over the 3D MNI coordinates, yielding a unified, electrodes location-aware representation:

1. Coordinate Normalisation For each electrode i we take its MNI position $\mathbf{p}_i = (x_i, y_i, z_i)$ and rescale every axis to $[-1, 1]$.

2. 3-D Fourier Positional Embedding We use $n_f = 16$ harmonics per axis. For every triplet $(k, \ell, m) \in \{0, \ldots, n_f - 1\}^3$ we compute

$$\cos 2\pi(kx_i + \ell y_i + mz_i), \sin 2\pi(kx_i + \ell y_i + mz_i).$$

Concatenating the cosine and sine terms over all n_f^3 triplets yields an embedding vector $\mathbf{e}_i \in \mathbb{R}^{d_{\text{pos}}}$ with

$$d_{\text{pos}} = 2\, n_f^3 = 2 \times 16^3 = 8\,192.$$

3. Attention Weights Each output dimension $j = 1, \ldots, D_1 (D_1 = 270)$ has a learnable query vector $\mathbf{q}_j \in \mathbb{R}^{d_{\text{pos}}}$. Channel weights are

$$\alpha_{j,i} = \frac{\exp\left(\mathbf{q}_j^\top \mathbf{e}_i\right)}{\sum_{k=1}^{C} \exp\left(\mathbf{q}_j^\top \mathbf{e}_k\right)}. \tag{5.4}$$

4. Weighted Aggregation Given the time–channel input $X \in \mathbb{R}^{T \times C}$, the FSA feature for dimension j is

$$\text{FSA}_{\text{3D}}(X)^{(j)} = \sum_{i=1}^{C} \alpha_{j,i}\, X^{(i)}, \qquad j = 1, \ldots, 270.$$

5. Spatial Dropout and 1×1 Projection During training we randomly mask electrodes within a radius $d_{\text{drop}} = 0.2$ in MNI space to foster robustness. A 1×1 convolution then maps the 270-D spatial vector to the feature dimension expected by the temporal decoder.

sEEG Decoder Variants

All decoder variants can operate on raw sEEG or on FSA outputs, followed by strided and transposed 1D convolutions to align to 50 Hz.

Recurrent and Hybrid Models

- **Baseline GRU:** 1D temporal convolution + three bidirectional GRU layers, then transpose-conv to 50 Hz (100 logits for aligned CE) or 125 Hz (101 logits for CTC).
- **Channel-Attention + RNN:** With spatial attention to aggregate features in the channel dimension, then RNN temporal integration.

Transformer-Based Models

- **Temporal Transformer:** Self-attention over time treating electrodes as features.
- **Transformer + [CLS] Token:** Adds a learnable classification token for HuBERT-unit decoding.
- **SwinTW** [21]: Windowed self-attention adapted from Swin Transformer for time series, described in Chapter. 4 in detail.

- **iTransformer** [94]: Embeds each channel's time series as a separate variate token with instance normalization, capturing inter-channel correlations via cross-token attention and intra-channel dynamics via a shared FFN.
- **PatchTST** [95]: Segments each channel into non-overlapping temporal "patches" treated as tokens, sharing a unified Transformer backbone across channels and optionally pretrained with a masked-patch objective.

Alignment Strategies

- *Aligned decoding:* Frame-wise MSE (articulatory) or cross-entropy/Focal loss (HuBERT units).
- *Non-aligned (CTC) decoding:* CTC loss on HuBERT unit sequences to accommodate timing variability.

Articulatory-space targets use only the aligned regime; HuBERT-unit decoding supports aligned and non-aligned training.

5.2.7 Experimental Settings

We systematically evaluate our sEEG-to-speech decoding pipeline by varying three factors—decoder architecture, loss function, and input feature set—and by comparing strict versus non-strict cross-validation schemes.

Model Variants & Loss Functions We test the following decoder architectures (each followed by strided/transposed convolutions to 50 Hz):

- **Recurrent Models:** Bidirectional GRU baseline; Channel-Attention + RNN.
- **Transformer-Based Models:** Temporal Transformer; Transformer + [CLS] token; SwinTW [21]; iTransformer [94]; PatchTST [95].

- **Spatial Attention Head:** The Fourier Spatial Attention (FSA) head can be prepended to any of the above architectures except SwinTW and Channel-Attention RNN.

For HuBERT-unit decoding we compare three classification losses: aligned cross-entropy, focal loss [92], and CTC loss [88]. For articulatory targets we use aligned regression losses: MSE and Huber [93].

Input Feature Sets We consider two frequency-band configurations of the raw sEEG:

- **High-Gamma only:** 70–150 Hz band-pass filtered envelopes.
- **High-Gamma + Low-Frequency:** concatenation of the 70–150 Hz envelope with 1–30 Hz band-pass filtered signals.

Cross-Validation Protocols

- *Strict CV:* Words in test set are *unseen* during training (zero overlap in vocabulary).
- *Non-Strict CV:* Train/test split by trial only, allowing shared words between sets.

All experiments report results under both protocols to assess decoder generalization to novel vocabulary versus familiar contexts.

5.2.8 Evaluation metrics

Instead of using PCC and STOI+, as in Chapters 3 and 4, we evaluate model performance using decoding accuracy and edit distance. One key reason for not employing alignment-based metrics such as PCC and STOI+ is that the

Tacotron2 model (used in the HuBERT framework) resynthesizes the mel spectrogram, which does not necessarily match the temporal length of the original speech's mel spectrogram. As a result, alignment-dependent evaluation metrics, including PCC, STOI+, and MCD, are not applicable in this context. We use edit distance and word matching accuracy described in subsection. 2.2.2.

5.3 Results

We apply the sEEG speech decoding framework to 11 sEEG-only participants. Perceptually better-decoded speech is found compared to ECoG-decoded speech using the synthesizer described in Chapter 3. An example decoded speech can be found on the web page: (`https://xc1490.github.io/sEEG/`). We evaluate the performance of our sEEG decoding model with the abovementioned evaluation metrics.

Fig. 5.4 presents box-plots of all evaluation metrics for the 11 participants when HuBERT units are used as the latent representation. The neural decoder is an RNN topped with a Fourier Spatial Attention (FSA) head. Results are obtained under the *non-strict* cross-validation protocol (training and test sets may share vocabulary). The decoding and resynthesis accuracy represents the decoding accuracy of the transcribed word from the resynthesized and decoded speech. The error could be due to the imperfect speech decoding and the ASR speech-to-text process(ASR models perform worse in single-word recognition as they lack the contextual information). We could observe that most participants had an accuracy of around 0.1, and the best participant got a decoding accuracy of 0.5, which is unheard of. We also report a *time-lagged spectrogram correlation*. Log-magnitude spectrograms of the decoded and reference speeches

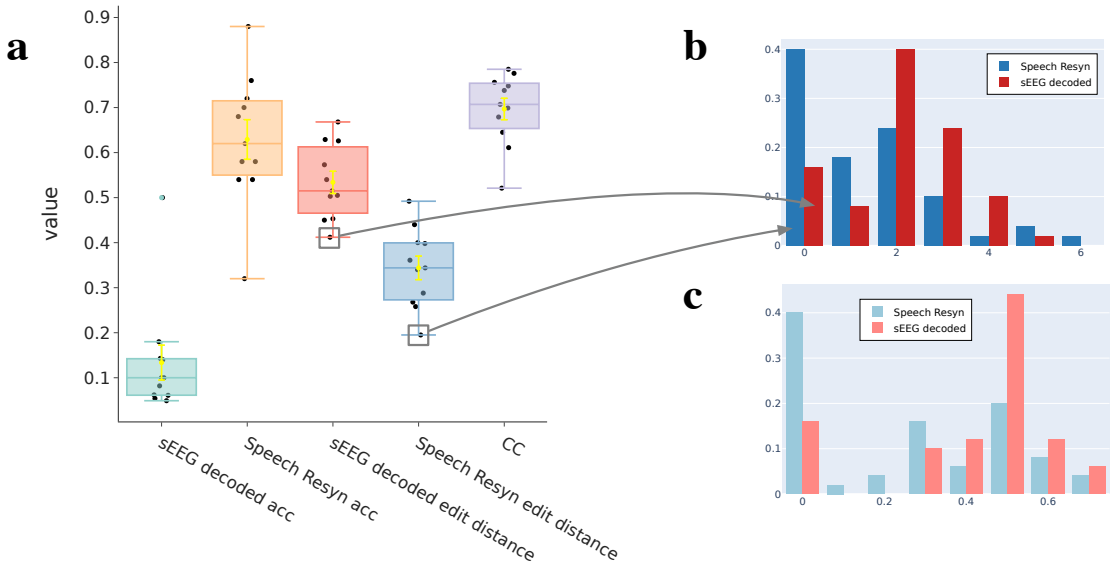


FIGURE 5.4: | **sEEG speech decoding performance with HuBERT representation** Here, we used the decoding accuracy to evaluate the performance of our neural speech decoding pipeline. The neural decoder is an RNN topped with a Fourier Spatial Attention (FSA) head. Results are obtained under the *non-strict* cross-validation protocol. A pre-trained HuBERTCTC [17] model automatically generates transcripts from the decoded and resynthesized speech. We then evaluate the percentage of correctly decoded words. CC is the peak Pearson correlation across all frame offsets between the reference and decoded log-spectrograms. The Levenshtein distance [96] is used to evaluate the edit distance between the decoded/resynthesized words and the ground truth word. We also show in **b** and **c** the detailed distribution of the Levenshtein distance of each trial (original and normalized) of the best performing participant.

are cross-correlated over every possible frame offset, and the peak Pearson coefficient is taken as the score (last column of Fig. 5.4). We also evaluate the edit distance on the resynthesized and decoded speech. The lower the edit distance, the better the decoding. We also notice that much of the decoded speech is partially correct, meaning that some phonemes are correct. This suggests that the sEEG decoder is working at the phoneme level. We also found that our model can generalize to unseen words from perceptual listening. We hypothesize that this is because the model can generate the phoneme-level information and the transition between phonemes based on what it has learned.

(A) HuBERT-Based Decoding

Model	Loss	CV	e2a Acc.	a2a Acc.	PCC	Unit Acc.
Temporal Transformer	Focal	Non-strict	0.50	0.72	0.81	0.38
FSA RNN	Focal	Non-strict	0.48	0.72	0.80	0.40
iTransformer	Focal	Non-strict	0.48	0.72	0.80	0.40
FSA RNN	CE	Non-strict	0.46	0.66	0.80	0.39
FSA RNN	Focal	Strict	0.12	0.70	0.67	0.18
Temporal Transformer	Focal	Strict	0.12	0.74	0.66	0.15

(B) Articulatory Space based Decoding

Model	Loss	CV	e2a Acc.	a2a Acc.	PCC	Art PCC
FSA RNN	MSE+Huber	Non-strict	0.42	0.74	0.81	0.79
Temporal Transformer	MSE+Huber	Non-strict	0.42	0.72	0.82	0.80
FSA RNN	Huber	Non-strict	0.42	0.68	0.81	0.78
PatchTST	MSE+Huber	Non-strict	0.40	0.64	0.73	0.77
Temporal Transformer	Huber	Strict	0.14	0.74	0.73	0.70
FSA RNN	MSE+Huber	Strict	0.12	0.68	0.69	0.68

TABLE 5.2: Decoding performance for one example subject under non-strict and strict cross-validation. *e2a* and *a2a* denote sEEG-to-speech and re-synthesis accuracies; PCC is the Pearson correlation coefficient between decoded and ground-truth feature trajectories; "Unit Acc." refers to HuBERT unit classification accuracy, and "Art PCC" means the correlation between decoded and reference articulatory space.

We systematically evaluated decoding performance across multiple architectures and loss functions on the best-performing participant (Table 5.2). For HuBERT-based decoding, focal loss consistently yielded higher e2a accuracy compared to cross-entropy loss, suggesting its suitability for handling the inherent class imbalance in neural-to-unit mappings. In the articulatory space framework, combining MSE and Huber losses outperformed the use of Huber loss alone in non-sctrict cross-validation, highlighting the benefit of multi-objective regression strategies for articulatory feature prediction.

It is important to note that the a2a accuracy reported in Table 5.2 reflects the performance of the speech-to-speech resynthesis pipeline, independent of the neural decoding process, and thus serves as a reference for the upper bound of achievable performance.

Across both representation spaces, we observed that models based on recurrent neural networks (RNNs) and temporal transformers consistently performed well, and incorporating the FSA head also shows benefits. Additionally, performance consistently declined under strict cross-validation, where test words were excluded from training, underscoring the challenge of generalizing to unseen lexical items in single-word decoding tasks with limited data. Notably, articulatory space models demonstrated competitive or even superior generalization under strict CV compared to HuBERT models, suggesting stronger inductive biases that support cross-word generalization.

In Fig. 5.5, we show some example speech spectrograms of the best-performing participant with HuBERT representations. More examples can be found at `https://xc1490.github.io/sEEG/`. Compared to the results shown in Chapter. 3, the decoded speech is closer to the resynthesized speech. We hypothesize that this is due to the more powerful HuBERT synthesizer and the simplification of the sEEG to latent tasks.

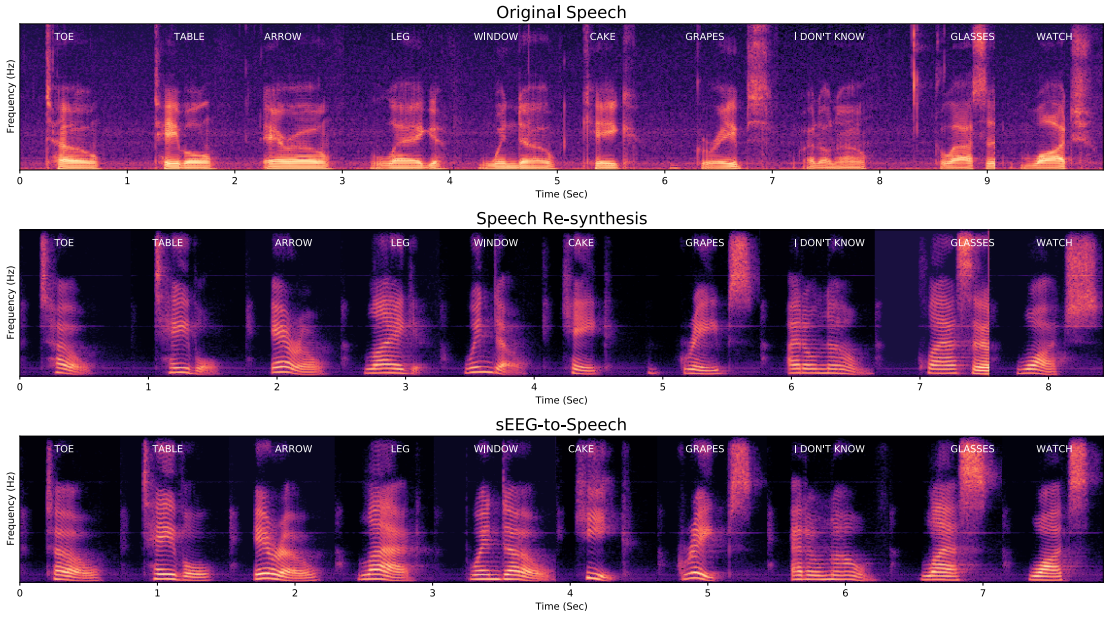


FIGURE 5.5: **Overt speech decoding via HuBERT representation using sEEG electrodes only.** Shown are example spectrograms from a single participant: (a) the original speech signal; (b) resynthesized speech produced by the HuBERT synthesizer (trained with CTC loss, which permits temporal misalignment between input and output); (c) decoded speech obtained by passing the discrete units predicted from sEEG through Tacotron 2 and WaveGlow. For the sEEG decoder, we employ a focal loss to align predicted units with those generated by the HuBERT model under an assumed frame-level correspondence between neural activity and acoustic frames. Remarkably, the sEEG-based decoding achieves precise detection of speech onset and offset.

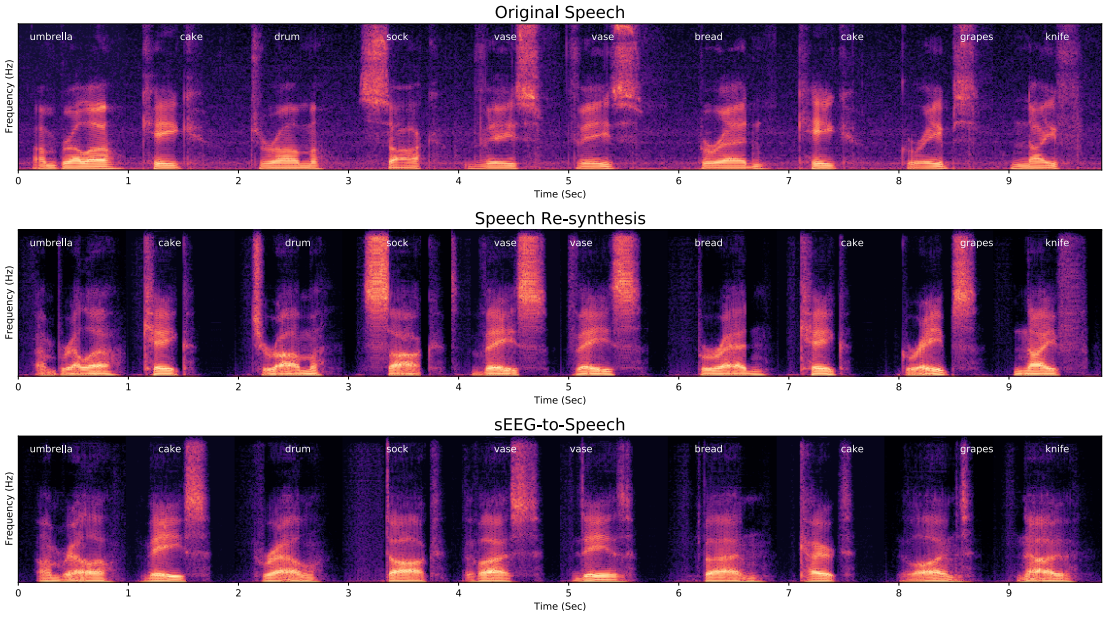


FIGURE 5.6: **Overt speech decoding via articulatory-space representations from sEEG electrodes only.** Example spectrograms from a single participant are shown: (a) the original speech signal; (b) resynthesized speech produced by the Articulatory synthesizer (HiFiGAN which takes the articulatory space and speaker embedding); (c) decoded speech obtained by mapping sEEG-derived articulatory features through HiFiGAN. The sEEG decoder is optimized with a MSE + Huber loss to align predicted articulatory space with those extracted from the audio, under an assumed frame-level correspondence between neural activity and acoustic frames. Importantly, the articulatory-space-guided decoding maintains precise detection of speech onset and offset.

In Fig.5.6, we also present example speech spectrograms from the best-performing participant using articulatory space representations. Additional examples are available at `https://xc1490.github.io/sEEG/`. Compared to the results in Fig.5.5, the decoded speech not only closely matches the resynthesized speech but also approximates the original speech, benefiting from the speaker-dependent characteristics of the articulatory space–based synthesizer.

5.4 Conclusions and Discussions

This chapter investigated sEEG-based speech decoding using two complementary representational frameworks: the HuBERT latent unit space and the articulatory space. By leveraging state-of-the-art self-supervised pretrained models, we systematically evaluated their capacity to decode overt speech from intracranial recordings.

Our results demonstrate that the HuBERT-guided decoding framework, which maps sEEG signals to quantized linguistic units, effectively leverages pretrained linguistic priors to generate intelligible speech, even under limited and noisy neural data conditions. This corroborates recent findings on the linear correspondence between cortical activity and representations in large speech models such as HuBERT and wav2vec 2.0. Notably, simpler RNN-based decoders outperformed more complex architectures like SwinTW, suggesting that under discrete prediction paradigms, smaller models can achieve robust performance with constrained sEEG datasets.

The articulatory space framework provided an interpretable and physiologically grounded alternative. By regressing sEEG signals onto the spatiotemporal dynamics of articulatory movements and conditioning a HiFi-GAN vocoder, we

achieved speaker-specific, high-fidelity speech synthesis. Importantly, like HuBERT, this approach leverages large-scale pretrained models such as WavLM for speech feature extraction and HiFi-GAN for speech synthesis. Compared to the source-filter synthesizer introduced in Chapter 3, which was trained from scratch, the articulatory space framework offers superior stability and generalization across speakers, while preserving speaker-specific characteristics, highlighting its potential for clinically deployable and speaker-adaptive speech prostheses.

For evaluation, we employed decoding accuracy, edit distance, moving beyond alignment-dependent measures such as PCC or STOI+. Under the non-strict cross-validation setting, the best participants reached a word-level decoding accuracy of 50%, far surpassing the prior source–filter speech synthesizer. Importantly, even partially correct phoneme sequences yielded perceptually recognizable outputs, suggesting that the decoders successfully capture subword-level neural dynamics.

Despite these advances, our approach currently falls short of the WER benchmarks reported in [14], primarily due to differences in data scale. Whereas [14] utilized tens of hours of sentence-level recordings from a single participant, our dataset comprised merely five minutes per subject, limited to a 50-word vocabulary. This data constraint hampers both generalization and the effectiveness of advanced transformer-based decoders. Moving forward, expanding to more extensive sentence-level datasets will not only improve contextual modeling but also boost ASR evaluation performance, as word-level ASR tasks remain particularly challenging and lead to decreased performance in evaluation. Encouragingly, we have access to 10–20 hours of naturalistic speech data from hospitalized epilepsy patients with ECoG/sEEG implants, providing a promising avenue for future large-scale training.

We have several avenues for further exploration. First, decoding imagined speech represents a critical challenge, where the absence of precise temporal markers will require alignment-free loss functions such as CTC or differentiable dynamic time warping loss. Second, gradient- and attention-based analyses will be essential to elucidate the contribution of specific electrodes and brain regions, particularly given the unique anatomical reach of sEEG, including hippocampal and white matter areas. Finally, comparative studies across overt, perceptual, and imagined speech will be instrumental in disentangling the shared and distinct neural mechanisms of speech processing, paving the way toward generalizable and robust neural speech prostheses.

In summary, our results show that uniting self-supervised speech resynthesis, learned speech representations, articulatory modeling, and neural decoding offers a powerful path forward for next-generation speech BCIs. By systematically evaluating HuBERT and articulatory space frameworks, we provide insights into their relative advantages and set the stage for future innovations in decoding naturalistic communication from the human brain.

Chapter 6

Phoneme Classification as Auxiliary Supervision and sEMG-to-Speech Decoding

6.1 Introduction

In the preceding chapters, we systematically investigated neural speech decoding across single- and multi-subject paradigms, leveraging both interpretable acoustic modeling and advances in self-supervised speech representations. Chapter 3 introduced a novel neural speech decoding framework based on a differentiable source-filter speech synthesizer, achieving unprecedented levels of realism by mapping ECoG signals to continuous speech parameters. Chapter 4 extended this framework to cross-subject generalization by developing the SwinTW architecture, demonstrating robust performance across seen and unseen subjects when trained on multi-participant data. Chapter 5 further advanced decoding performance by integrating state-of-the-art self-supervised speech models, such as HuBERT and WavLM (to map to articulatory space), to bridge

the mapping between neural signals and speech representations, and by exploring an articulatory space framework that enables speaker-specific synthesis with high fidelity.

In this chapter, we pursue two complementary directions to improve neural speech decoding further and expand its clinical applicability. First, we investigate the utility of an auxiliary phoneme classification objective to regularize articulatory-space decoding, hypothesizing that phoneme supervision can sharpen phonemic transitions and improve the robustness of articulatory decoding. To this end, we have already trained a phoneme classifier that operates on articulatory features extracted by a pretrained articulatory feature encoder; embedding this classifier into the end-to-end neural decoder remains an important avenue for future work. Second, we explore non-invasive decoding using surface electromyography (sEMG), which captures articulatory muscle activity and offers a promising pathway toward scalable speech interfaces. By leveraging the articulatory space as a physiologically grounded intermediate representation, we assess the feasibility of sentence-level speech decoding from sEMG signals, laying the foundation for future work on non-invasive speech neuroprostheses for individuals with severe motor speech impairments.

6.2 Auxiliary Phoneme Classification in Articulatory Space

To further improve decoding of the 14-dimensional articulatory representation from neural or sEMG data, we add an auxiliary phoneme-classification objective. Prior studies have shown that both self-supervised speech features (e.g.,

HuBERT) and articulatory trajectories are strongly correlated with phoneme identity [97, 98].

Current scope. Here we *pre-train* a phoneme classifier using only speech audio. Waveforms are first passed through a frozen articulatory-feature extractor; a separate network is then trained to map these features to phoneme labels. Neural recordings are **not** involved at this stage. The resulting classifier encodes phonemic decision boundaries in articulatory space and is intended, in future work, to supply an additional loss term during neural-to-articulatory training—by freezing its weights and back-propagating a cross-entropy, focal, or CTC loss. This section details the pre-training procedure and discusses how the classifier can benefit downstream neural speech decoding.

By explicitly requiring the 14-dimensional decoder outputs to drive a frame-wise 39-way phoneme classifier (at 50 Hz), we inject discrete, segmental supervision that:

- *Align representations with phoneme-relevant articulatory gestures,* such as lip closure versus release or tongue constriction patterns, so each decoded dimension captures linguistically meaningful motion features.
- *Counters the over-smoothing bias* of pure regression losses, which tend to blur rapid transitions: the classification objective forces the model to produce sharp vector changes at phoneme onsets and offsets.
- *Enhances robustness and generalization* by combining continuous trajectory reconstruction with a structured phoneme signal, reducing overfitting to any single corpus.

The complete articulatory to phoneme classification pipeline is illustrated in Fig. 6.1.

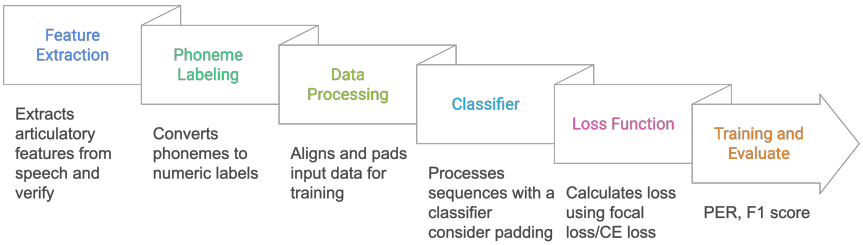


FIGURE 6.1: An end-to-end Articulatory to Phoneme Workflow: (1) *Feature Extraction* uses an articulatory-space encoder to produce 14-dimensional features at 50 Hz; (2) *Phoneme Labeling* converts aligned boundaries into 39 classes; (3) *Data Processing* trims, pads (with label -100) and records true lengths for packed RNN/CTC; (4) *Classifier* (BiLSTM or Conformer) outputs frame-wise logits; (5) *Loss Function* applies auxiliary objectives (CTC, cross-entropy, focal, mixup, KL) on non-padding frames; (6) *Training & Evaluation* optimizes classification and reports frame accuracy, phoneme error rate (PER) and F1 scores.

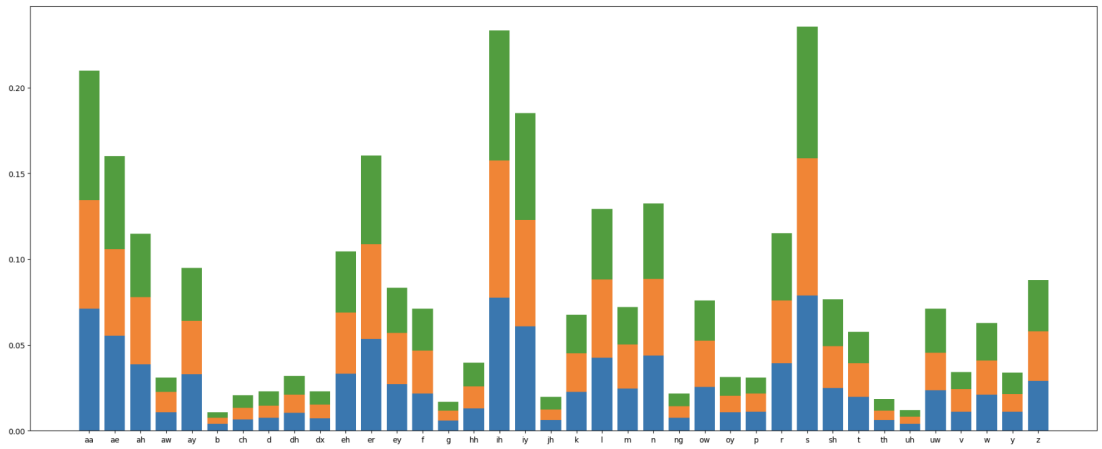
6.2.1 Phoneme Datasets and Preprocessing

We use two corpora with time-aligned phoneme labels, remapped to 39 classes (including `h#` for silence) and sampled at 50 Hz:

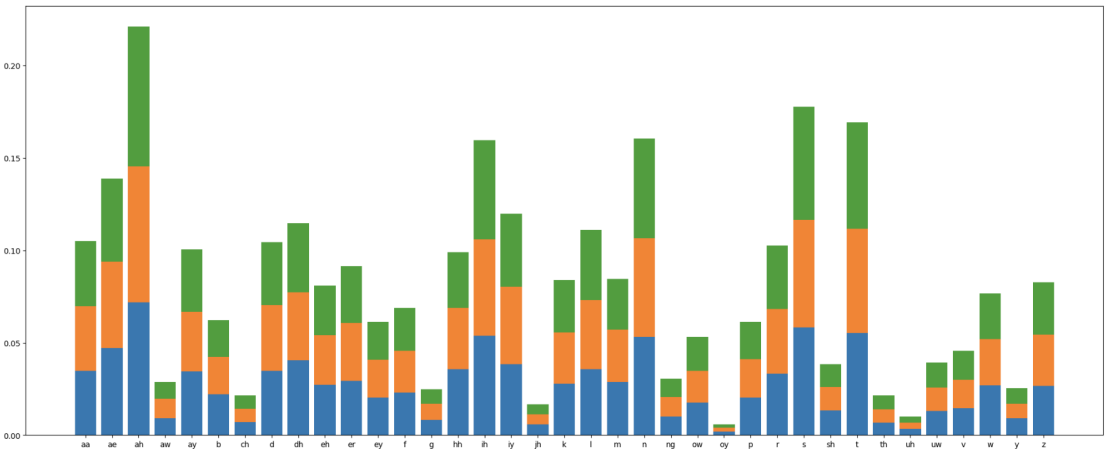
- **TIMIT (MS-WAV)**[1] Train: 137 min, Valid: 17 min, Test: 17 min. Original 61 symbols to 39 classes; converted to frame labels.
- **LibriSpeech Alignments** Train: 6035 min, Valid: 323 min, Test: 324 min. We map 61 ARPAbet classes $\rightarrow$ 39 CMUdict classes.

Padding frames use label -100; silence (`h#`) is class 15. Fig. 6.2 illustrates the pronounced class imbalance in phoneme frequencies for both TIMIT and LibriSpeech, where a small subset of phonemes accounts for most frames.

[1] LDC Catalog LDC93S1



(A) TIMIT



(B) LibriSpeech

FIGURE 6.2: Frame-wise phoneme frequency distributions for TIMIT (top) and LibriSpeech (bottom), with training (blue), validation (orange) and test (green) overlaid. Both corpora exhibit pronounced class imbalance: a few phonemes (e.g. /ah/, /ae/, /ih/, /iy/, /s/) account for most frames, while many others are rare. The similar patterns across train/val/test indicate consistent imbalance across splits.

6.2.2 Data Collation and Batching

- **Trim & Pad:** For each batch of size B, trim all sequences to the minimum length $L_{\min}$, then pad features to $(B, L_{\max}, 14)$ with zeros and labels to $(B, L_{\max})$ with -100.
- **Packed Lengths:** Record true lengths $\{\ell_i\}_{i=1}^{B}$ (frames $\neq -100$) for RNN/CTC packing.

6.2.3 Model Architectures for the Phoneme Classifier

Auxiliary Phoneme Classifier (Stage 1 Pre-training) We compare two lightweight architectures:

1. **3-layer BiLSTM:** input $\rightarrow$ LSTM$(14 \rightarrow 128, \times 3, \text{bi}, \text{dropout } p = 0.1) \rightarrow$ Linear$(256 \rightarrow 39)$.
2. **3-layer Conformer:** dim $= 256$, heads $= 8$, conv_kernel $= 5$, dropout $p = 0.1$, followed by Linear$(256 \rightarrow 39)$.

Train on $\{\mathbf{a}_{\text{audio}}(t), y_{\text{phn}}(t)\}$ with cross-entropy until convergence; then freeze classifier weights. $\mathbf{a}_{\text{audio}}(t)$ is 14-D articulatory feature from audio as input. $y_{\text{phn}}(t)$ is ground truth phoneme label. Both were sampled at a 50 Hz frame rate.

6.2.4 Loss Functions

Let T be the number of decoded frames and $C = 39$ the phoneme classes. Denote:

$$p_k(t) = \text{softmax}(\ell\text{ogits}(t))_k.$$

If phoneme labels are available, we consider the following auxiliary loss options:

1. Cross-Entropy Loss:

$$\mathcal{L}_{\text{CE}} = \frac{1}{T} \sum_{\substack{t \\ y_{\text{phn}}(t) \neq -100}} -\log p_{y_{\text{phn}}(t)}(t),$$

where $p_y(t)$ denotes the predicted probability for phoneme y at time t.

2. Focal Loss:

$$\mathcal{L}_{\text{foc}} = \frac{1}{T} \sum_{\substack{t \\ y_{\text{phn}}(t) \neq -100}} \left(1 - p_{y_{\text{phn}}(t)}(t)\right)^{\gamma} \cdot \left[-\log p_{y_{\text{phn}}(t)}(t)\right],$$

which down-weights well-classified phonemes and focuses on hard examples.

3. CTC Loss:

$$\mathcal{L}_{\text{CTC}} = -\frac{1}{T} \log p_{\text{CTC}}\left(y_{\text{phn}} \mid \text{logits}_{1:T}\right),$$

where p_{CTC} denotes the conditional probability of the phoneme sequence under the CTC decoding alignment.

4. Frame-Aligned Mixup (KL Loss). For every time index t we pick two utterances u_i, u_j in the mini-batch and take their articulatory frames $\mathbf{x}_{i,t}, \mathbf{x}_{j,t}$ with one-hot labels $y_{i,t}, y_{j,t} \in \{0,1\}^P$ (P phoneme classes). Drawing $\lambda \sim \text{Beta}(\alpha, \alpha)$, we create

$$\tilde{\mathbf{x}}_t = \lambda\, \mathbf{x}_{i,t} + (1-\lambda)\, \mathbf{x}_{j,t}, \qquad \tilde{y}_t = \lambda\, y_{i,t} + (1-\lambda)\, y_{j,t}.$$

Because $\tilde{y}_t$ is now a *soft* label, the loss is defined as a Kullback–Leibler divergence between this target distribution and the model prediction:

$$\mathcal{L}_{\text{mixup}} = \mathbb{E}_{t,\lambda} D_{\text{KL}} \left(\tilde{y}_t \| \operatorname{softmax} \left(\tilde{\ell}_t \right) \right), \quad \tilde{\ell}_t = \lambda \ell_{i,t} + (1-\lambda)\ell_{j,t}. \tag{6.1}$$

Here $D_{\text{KL}}(p\|q) = \sum_p p \log \frac{p}{q}$ and $\ell_{i,t}$ (resp. $\ell_{j,t}$) are the logits before softmax for frame t of utterance u_i (resp. u_j). This KL formulation properly handles the soft targets produced by mixup while promoting smoother phoneme boundaries and greater robustness.

5. KL–Prior Loss. Instead of supervising each phoneme with a one-hot target, we encode prior knowledge about phoneme confusability. Let $\mathbf{C} \in \mathbb{R}^{P \times P}$ be the phoneme-confusion matrix obtained from our best phoneme classifier and row-normalized so that $\sum_k C_{y,k} = 1$ for every phoneme y. Row y therefore gives a soft label $\text{softConfusion}_y := (C_{y,1}, \ldots, C_{y,P})$ that reflects how often phoneme y is mis-recognized as each other class.

For every valid frame t ($y_{\text{phn}}(t) \neq -100$) we compute the Kullback–Leibler divergence between this prior and the model's predicted distribution $p_\cdot(t) = \text{softmax}(\ell(t))$:

$$\mathcal{L}_{\text{KL}} = \frac{1}{T} \sum_{\substack{t \\ y_{\text{phn}}(t) \neq -100}} D_{\text{KL}}$$

$$\left(\text{softConfusion}_{y_{\text{phn}}(t)} \| p_\cdot(t) \right). \tag{6.2}$$

This loss encourages the network to allocate residual probability mass to the phonemes that are *a-priori* most confusable with the ground-truth class, thereby providing a smoother and more informative supervisory signal than a strict one-hot target.

Each loss was evaluated separately during development. We found that the $\mathcal{L}_{\text{mixup}}$ auxiliary objective yielded the best performance in phoneme decoding and thus adopted it as our primary phoneme loss.

6.2.5 Evaluation Metrics for Phoneme Classification from Articulatory Space

- **Frame Accuracy**: excluding padding (-100), with/without silence (`h#`).
- **Phoneme Error Rate (PER)**: $1 - \text{accuracy}_{\text{no-silence}}$.
- **Macro-F1** and **Weighted-F1** over 39 classes.
- **Confusion Matrix**: computed on valid (non-padding, non-silence) frames.

6.2.6 Articulatory to Phoneme Classification Results

As shown in Tables 6.1 and 6.2, the LSTM classifier plus Mixup loss consistently outperforms all other configurations on both TIMIT and LibriSpeech, achieving the highest frame accuracy and lowest phoneme error rate. Conformer variants yield slightly lower performance across the same losses, and purely CTC- or focal-based objectives are less effective than mixup. These results confirm that mixup is the most beneficial auxiliary loss for articulatory-to-phoneme pretraining, particularly when paired with an LSTM classifier.

6.2.7 Analysis and Plan

Fig. 6.3 shows that the articulatory-to-phoneme decoder achieves high accuracy on the diagonal but systematically confuses phonetically similar categories (e.g.

TABLE 6.1: TIMIT articulatory to phoneme pretraining results

Loss	Model	Frame Accuracy (%)	PER (%)	Macro F1 (%)
Mixup	LSTM	80.58	14.89	75.98
CTC and Focal	LSTM	78.80	16.37	73.35
Soft-Confusion and Focal	LSTM	79.65	16.89	74.54
Focal only	LSTM	78.62	17.79	73.37
Mixup	Conformer	80.05	18.93	74.47
CTC and Focal	Conformer	77.89	18.39	72.48
Soft-Confusion and Focal	Conformer	78.12	18.76	72.63
Focal only	Conformer	79.78	18.38	74.59

TABLE 6.2: LibriSpeech articulatory to phoneme pretraining results

Loss	Model	Frame Accuracy (%)	PER (%)	Macro F1 (%)
Mixup	LSTM	85.06	8.74	84.08
CTC and Focal	LSTM	84.58	8.89	81.01
Soft-Confusion and Focal	LSTM	85.43	8.97	84.28
Focal only	LSTM	84.63	9.57	83.34
Mixup	Conformer	84.49	10.41	83.15
CTC and Focal	Conformer	83.29	12.18	79.91
Soft-Confusion and Focal	Conformer	82.76	12.41	79.32
Focal only	Conformer	82.92	13.51	81.62

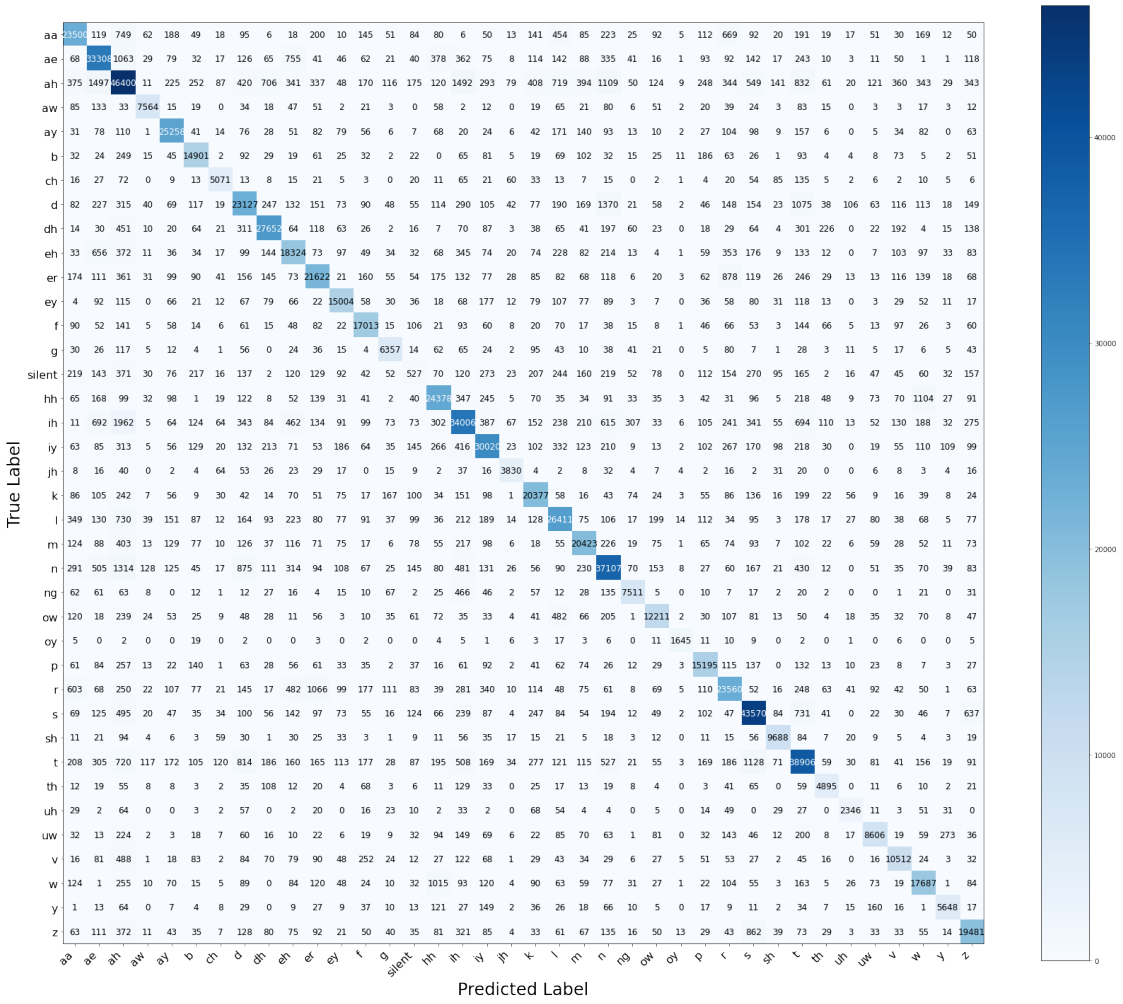


FIGURE 6.3: Confusion matrix for the best-performing articulatory→phoneme decoder on TIMIT. Rows correspond to true labels and columns to predicted labels; cell values are counts (darker shading = more confusion). Common errors occur between acoustically similar pairs (e.g. /ih/→/ah/, /er/→/r/, /d/→/n/, and silence.

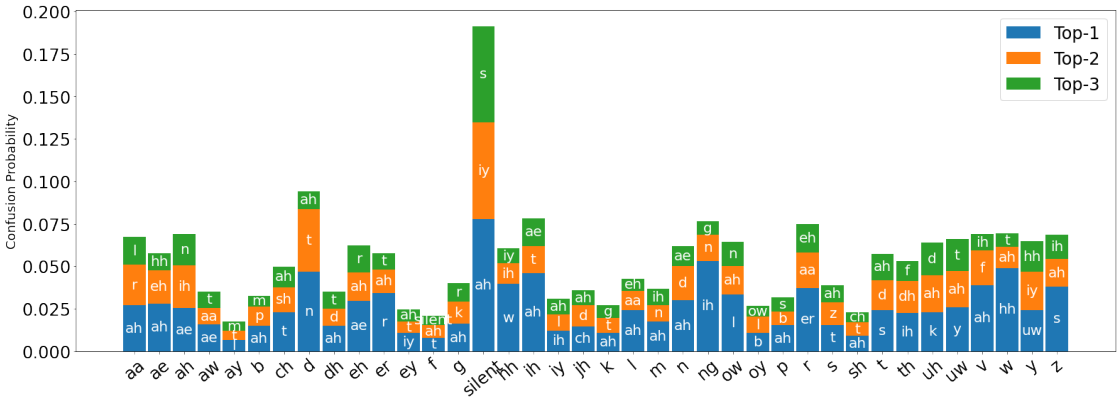


FIGURE 6.4: Top-3 confusion probabilities for each of the 39 phoneme classes (blue = Top-1, orange = Top-2, green = Top-3). For each true class on the x-axis, the stacked bars show how often the most frequent incorrect predictions occur. The largest confusions are /ih/→/ah/, /v/→/ah/, /d/→/n/, and silence→/ah/, silence→/ih/, reflecting typical articulatory–acoustic overlaps.

voiced vs. voiceless stops, palatal vs. non-palatal fricatives). Fig. 6.4 further emphasizes these patterns and reveals that silent frames frequently appear among the top-3 misclassifications, indicating that the model's handling of silence remains a key area for improvement.

To recap, our initial articulatory to phoneme classifier achieves strong frame-level accuracy, but still struggles with under-represented silence and benefits from richer phoneme supervision. We therefore propose two enhancements:

1. **Separate Silence Detection.** Silence is rare yet frequently mistaken for vowels (e.g. /ih/). We will:
   - Add a binary *silence vs. non-silence* head trained with *focal loss* to counter class imbalance.
   - Use a two-stage classification setup: first detect silence, then route only non-silence frames into the 38-way phoneme classifier.
2. **Phoneme-Regularized Decoding.** Leveraging our pretrained articulatory→phoneme model, we will inject phoneme supervision into upstream decoding:

- In the *sEEG→articulatory→speech* pipeline, freeze the phoneme head and fine-tune the sEEG→articulatory encoder with an auxiliary phoneme loss.
- In the *sEMG→articulatory→speech* pipeline, similarly use the phoneme classifier to regularize the mapping, ensuring sEMG-derived articulatory features remain phoneme-predictive.

6.3 sEMG to speech decoding leveraging deep learning

6.3.1 introduction

Surface electromyography (sEMG) captures muscle activity from the skin surface and offers a promising non-invasive interface for speech decoding. In contrast, the majority of prior work on speech neuroprostheses has relied on invasive neural signals, such as electrocorticography (ECoG), which, despite their high performance, present significant barriers to clinical scalability. In contrast, sEMG can detect articulatory muscle activity in individuals who retain partial motor control of the face and neck, even without vocal output. This makes it a compelling modality for developing speech restoration technologies for patients with severe motor speech disorders, including anarthria and dysarthria, particularly when caused by conditions such as ALS, stroke, or Parkinson's disease.

Our central intuition is that sEMG signals—although noisy, speaker-specific, and lacking acoustic structure—nonetheless preserve sufficient articulatory dynamics to support speech decoding, especially when guided by a structured latent representation. Specifically, we posit that decoding performance can be

improved by introducing a physiologically grounded inductive bias: an intermediate articulatory space that aligns with interpretable speech-related motor trajectories.

In this section, we develop a deep learning-based sEMG-to-speech decoding framework that incorporates an articulatory latent space to improve both interpretability and cross-condition generalization. Our ultimate goal is to build a model that (1) generalizes across subjects, (2) transfers effectively from overt (aloud) speech to mimed and subvocal (imagined) speech, and (3) performs robustly on real patients with severe speech impairments, such as anarthria. In the following sections, we describe the design of our decoding framework in the aloud speech setting and present preliminary results that demonstrate its effectiveness on aloud speech decoding.

6.3.2 sEMG Data Collection

We collected surface electromyography (sEMG) data from 38 healthy, native English–speaking participants (no history of speech or neuromuscular disorders) and one patient with severe speech impairment (unable to articulate intelligible speech, producing only unclear vocalizations). The experimental design, data collection, and preprocessing were all carried out by Beatrice Fumagalli. The study was approved by the Institutional Review Board of NYU.

Tasks To evaluate speech decoding performance under different degrees of articulatory movement and vocalization, each participant performed three tasks:

- **Aloud**: Overt vocalized reading with normal speech production.
- **Mimed**: Silent articulation involving visible mouth movements without producing audible sound.

- **Sub-vocal**: Minimal to no muscle movement, but still has muscle activity detectable by surface electromyography (sEMG).

The fifty distinct sentences are drawn from the TIMIT corpus. In each condition, the set of fifty sentences was presented in three separate sessions (i.e., each sentence is repeated once per session), yielding 150 sentences per condition (Aloud/Mimed/Sub-vocal) per participant.

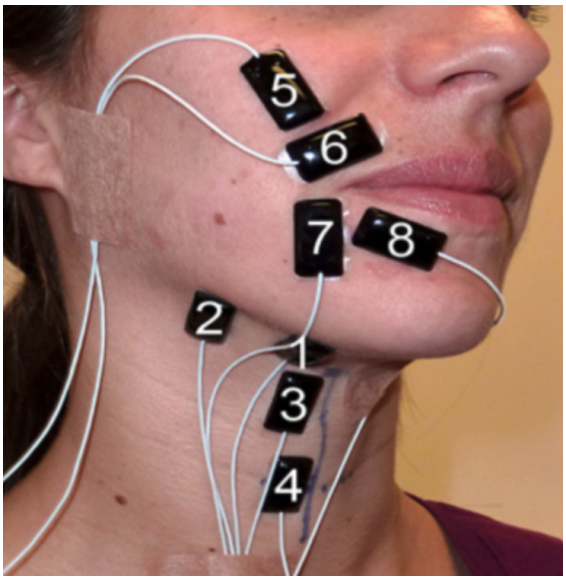


FIGURE 6.5: sEMG electrodes location

Eight surface electrodes were affixed to each participant's face and neck to capture the principal articulatory muscle activations (see Fig. 6.5). Electrode sites covered:

- **Submental region:** two electrodes beneath the chin over the suprahyoid muscles;
- **Masseter region:** two electrodes on the jawline over the masseter muscles;
- **Perioral region:** two electrodes around the lips over the orbicularis oris;

- **Buccal region:** two electrodes on the cheeks over the zygomaticus/risorius area.

The sEMG signals were recorded at 2 kHz using a bioamplifier, time-locked to the visual presentation of each sentence.

6.3.3 Signal Preprocessing

The signal preprocessing part is carried out by Beatrice Fumagalli. Raw sEMG was processed as follows (all operations at the original 2 kHz rate):

1. **DC Offset Removal.** Subtract the channel-wise mean over the entire block:

$$x_{\mathrm{dc}}[n] = x[n] - \frac{1}{N}\sum_{k=1}^{N} x[k].$$

2. **Filtering.**

   - 4th-order Butterworth notch filter at 60 Hz (and harmonics) to suppress line noise.
   - 4th-order Butterworth band-pass filter from 10 Hz to 450 Hz to isolate the physiological sEMG band.

3. **Envelope Extraction.** Let $x_{\mathrm{filt}}[n]$ denote the preprocessed sEMG signal after DC removal, notch filtering, and band-pass filtering. Compute the root mean square (RMS) envelope of the filtered sEMG signal using a sliding window:

$$e[n] = \sqrt{\frac{1}{L}\sum_{k=n-L/2}^{n+L/2} x_{\mathrm{filt}}^2[k]},$$

   where L is the window length in samples.

4. **Normalization.** Normalize each envelope channel to unit variance using the global standard deviation across all sessions:

$$e_{\text{norm}}[n] = \frac{e[n]}{\sigma_e}, \quad \sigma_e = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e[k] - \mu_e)^2}.$$

5. **Downsampling.** Decimate the normalized envelope to 1000 Hz with an anti-aliasing filter.

6.3.4 Additional Speech Feature Preparation

To augment the sEMG and audio inputs, we extracted two complementary feature streams:

1. **Word- and Phoneme-Level Alignment via MFA** We applied the Montreal Forced Aligner (MFA) to the 16 kHz audio and corresponding orthographic transcripts. MFA (using the English acoustic model and CMU pronunciation dictionary) produced TextGrids containing precise onset-offset times $\{t^w_{\text{on}}, t^w_{\text{off}}\}$ for each word w and $\{t^p_{\text{on}}, t^p_{\text{off}}\}$ for each phoneme p. These time stamps were used to segment both the sEMG envelope and audio feature sequences into linguistically meaningful units.

2. **Articulatory Space Representation.** A pretrained articulatory encoder $f_{\text{art}}(\cdot)$ was used to convert the aligned audio into a low-dimensional kinematic representation at 50 Hz. First, the audio waveform is converted into mel-spectrogram frames $\mathbf{S} = \{\mathbf{s}_t\}_{t=1}^{T}$. Then

$$\mathbf{a}_t = f_{\text{art}}(\mathbf{s}_t), \quad t = 1, \ldots, T,$$

yielding a trajectory $\{\mathbf{a}_t\}$ that encodes tongue, lip, and jaw movements.

6.3.5 sEMG Decoding Pipeline

To better illustrate our overall decoding framework, we present an overview in Fig. 6.6. The figure depicts the whole pipeline of decoding speech from surface electromyography (sEMG) signals with the guidance of an articulatory latent space. Participants engage in aloud, mimed, or sub-vocal speech tasks, during which sEMG signals are recorded and processed. The preprocessed sEMG signals are fed into a neural decoder that maps them into a 14-dimensional latent space. This space is supervised by an articulatory reference derived from the original audio recordings, which includes 12 articulatory dimensions and 2 source features (pitch and loudness). The decoded latent features are concatenated with a speaker embedding (pre-extracted from a reference utterance of the same speaker) and fed to a HiFi-GAN vocoder to generate the output speech waveform. The reference articulatory features are also used to generate re-synthesized audio. This design enables interpretable supervision and promotes intelligible and realistic decoding.

A detailed description of the end-to-end sEMG-to-speech pipeline is provided below:

1. **Preprocessing.** Each raw sEMG block undergoes:

   DC removal $\rightarrow$ Notch and band-pass filtering (10–450 Hz)
   $\rightarrow$ Envelope extraction $\rightarrow$ Normalization
   $\rightarrow$ Downsampling to 1 kHz or 200Hz.

2. **Speech Feature Extraction.** This part is described in subsection. 6.3.4

3. **Channel Exclusion.**

   - *All Channels:* Use all 8 electrodes.

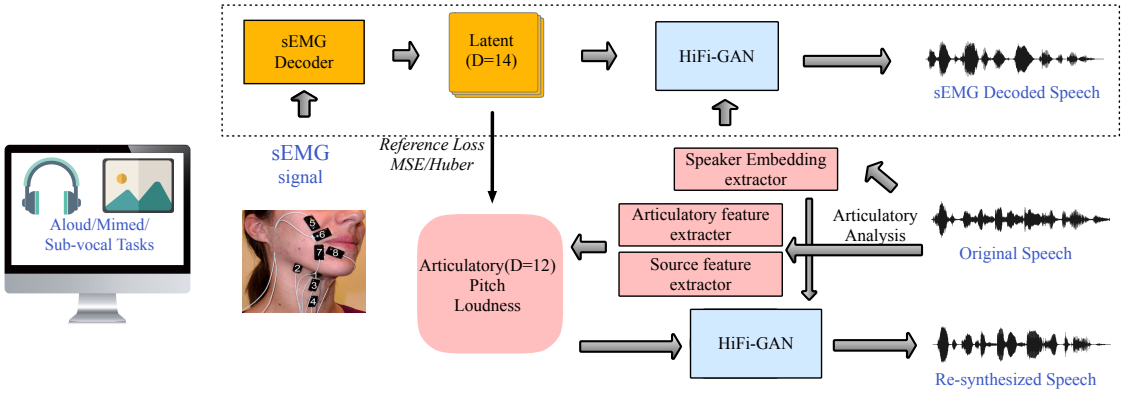


FIGURE 6.6: Participants perform aloud, mimed, or sub-vocal speech tasks, each involving three repetitions of 50 sentences selected from the TIMIT corpus. During these tasks, high-density surface electromyography (sEMG) signals are recorded and input into a neural sEMG decoder that maps them into a 14-dimensional latent representation. This latent space is then passed through a HiFi-GAN vocoder to synthesize speech. A reference articulatory space comprising 12 articulatory dimensions and two source features (pitch and loudness) is extracted from the original audio using a pipeline of source, articulatory, and speaker embedding extractors to supervise the learning of the latent space. The decoder is trained using reference loss (MSE or Huber) to align the predicted latent representation with the ground truth articulatory-source space. Re-synthesized speech is also generated from the reference features via HiFi-GAN for evaluation and comparison.

- *Exclude Ch. 8:* Drop the 8th channel (index 7) due to excessive noise and adhesion issues on the lower lip.

4. **Data Loader.**

   - *Padding:* pack each sentence to a fixed length $L_{\max}$ and store its true length ℓ_{orig}.
   - *Augmentation:* apply random additive noise and channel-masking to sEMG sequences during training.
   - *Mixup:* linearly interpolate pairs of sEMG and audio frames to regularize the model.

5. **Model Architectures.** We experiment with two backbone families for sEMG decoding:

   - *Conv+RNN:* convolutional front-end followed by a bidirectional LSTM stack.
   - *Conv+Transformer:* convolutional encoder feeding a causal or non-causal Transformer.

   All models accept 8 sEMG channels and optionally predict a 14-dimensional articulatory vector per frame. Padding masks are provided to inform attention/RNN layers of valid timesteps. A phoneme-classification auxiliary head may be added in future work.

6. **Loss Functions.** Let $\hat{\mathbf{a}}_t$ be the predicted articulatory vector and $\mathbf{a}_t$ the ground truth. We minimize

$$\mathcal{L}_{\text{art}} = \alpha \,\text{Huber}(\hat{\mathbf{a}}, \mathbf{a}) + (1-\alpha)\,\text{MSE}(\hat{\mathbf{a}}, \mathbf{a}),$$

with weight $\alpha \in [0, 1]$. We name *Huber MSE half loss* with setting $\alpha = 0.5$, and *Huber MSE soft loss* with setting $\alpha = 0.1$. An optional phoneme loss $\mathcal{L}_{\text{phn}}$ (cross-entropy) can be added on the auxiliary head.

7. **Evaluation.** For each test utterance, we:

   - Compute Pearson correlation coefficient (PCC) and mean squared error (MSE) between $\hat{\mathbf{a}}$ and $\mathbf{a}$.
   - Transcribe decoded audio using ASR model [30] and compute WER with ground truth transcript.

6.3.6 Results of sEMG decoding

Here we show some preliminary results obtained from two participants in the Table. 6.3

Model Selection and Evaluation. We conducted an extensive ablation study across various feature sets, loss functions, and preprocessing choices. For clarity, Table. 6.3 summarizes a subset of representative results, showing two selected configurations per participant. Our experiments use a non-strict cross-validation protocol, in which training and test sets may contain the same sentences recorded in different trials. All models share a Conv+RNN architecture, which consistently outperformed Conv+Transformer variants and was thus chosen for detailed analysis.

We experimented with two loss functions: *Huber soft* and *Huber half*, and observed minor performance variations across them. Word Error Rate (WER_{Pred}) is computed by transcribing the decoded waveform using a pretrained ASR model [30] and comparing it to the ground truth transcript. Phoneme Error Rate

(PER_{Pred}) is then computed based on the phoneme sequence of the transcribed words, providing a finer-grained assessment of pronunciation quality.

We also observe the exclusion of channel 8 generally improves decoding performance in P1, but this is not always true in P2.

Participant	Loss	Exclude	WER_{Pred}	PER_{Pred}	PCC_{art}	$STOI_{GT\text{-}Pred}$
P1	Huber soft	8-th chn	0.720	0.473	0.858	0.486
P1	Huber half	8-th chn	0.760	0.510	0.858	0.493
P2	Huber soft	8-th chn	0.854	0.698	0.739	0.364
P2	Huber soft	null	0.868	0.748	0.735	0.367

TABLE 6.3: Decoding performance metrics under different model configurations.

Demo Samples To qualitatively illustrate the effectiveness of our decoding framework, we provide a demo page containing representative audio samples: `https://xc1490.github.io/sEMG/`.

6.3.7 Future Work

Building upon the experimental framework and findings presented in this thesis, we outline several complementary directions to advance our sEMG-to-speech decoding research:

Single-Subject and Cross-Subject Modeling With data from 38 healthy participants, we will systematically compare subject-specific models, optimized for individual muscle activation patterns, with cross-subject models that generalize

across inter-speaker variability. Given the anatomically fixed electrode montage, cross-subject adaptation may be more tractable than intracranial modalities (e.g., ECoG, sEEG). Nevertheless, we will explore domain adaptation techniques, such as adversarial training, fine-tuning, or meta-learning, to mitigate residual inter-subject differences and improve generalizability.

Progressive Transfer from Aloud to Mimed and Sub-vocal Speech We propose a curriculum-based approach, beginning with *Aloud* speech, where sEMG-to-acoustic mapping is most direct. We will then fine-tune models on *Mimed* (silent) speech, using time-aligned pseudo-labels, and ultimately extend to *Sub-vocal* speech, where decoding relies solely on covert muscle activity. Addressing the low signal-to-noise ratio and weak ground truth in sub-vocal conditions will motivate using self-supervised or semi-supervised learning strategies.

Clinical Translation and Patient-Specific Adaptation In parallel, we will apply these models to data from a patient with severe speech impairment. We aim to transfer models trained on healthy speakers to patient data, first through fine-tuning and subsequently via domain-adaptive and personalized calibration approaches. This line of work will directly inform the development of real-time, patient-centered silent speech neuroprostheses.

Model Architecture Innovation We will continue benchmarking advanced neural architectures, including convolutional-recurrent hybrids, convolutional-transformer hybrids, and emerging generative models such as diffusion-based vocoders. Additionally, we plan to investigate graph-based and temporal attention models and lightweight architectures amenable to real-time deployment.

Optimal Feature Representation and Auxiliary Tasks We will expand our investigation of input representations, encompassing handcrafted features (e.g., RMS, TD-PSD, ZCR, MFCC), image-based spectrograms, and learned embeddings. Furthermore, we will explore the benefits of auxiliary tasks, such as phoneme classification, to encourage the model to preserve articulatory distinctions and enhance generalization.

Articulatory Space and Channel-Feature Correlation Analysis Finally, we will conduct detailed correlation and regression analyses between sEMG channels and articulatory or phonemic embeddings. This effort will inform decoder design by identifying the most informative electrodes and feature types and provide new insights into the mapping between surface muscle activity and articulatory representations.

Data Augmentation via Synthetic Proxy Speech A significant limitation of our current dataset is the scarcity of patient data, which includes only a single recording session and 50 sentences per condition (aloud, mimed, and subvocal). Moreover, we rely on proxy speech as the acoustic target for sEMG-to-speech decoding because the patient cannot produce intelligible vocalizations. To address this challenge, we propose leveraging zero-shot text-to-speech (TTS) models (e.g., E2 TTS [99]) to generate multiple synthetic speech renditions from the same transcript, varying pitch, loudness, speaking rate, and timbre. By pairing the same sEMG signal with diverse proxy speech outputs, we aim to create a rich augmented dataset that reflects both the invariant phonemic content and the variable acoustic realizations. This setup will allow the model to learn robust invariant phoneme embeddings while disentangling speaker-dependent

and prosodic variation, akin to the flow-matching [100] paradigm in generative modeling. Such an approach mitigates data scarcity and offers a promising direction for improving the generalization and adaptability of neural speech decoders.

Taken together, these directions aim to enhance the performance and generalization of silent speech decoding systems and advance our scientific understanding of the neuromuscular bases of speech, ultimately contributing toward clinically viable and user-centered communication interfaces.

Chapter 7

Conclusion and Future Work

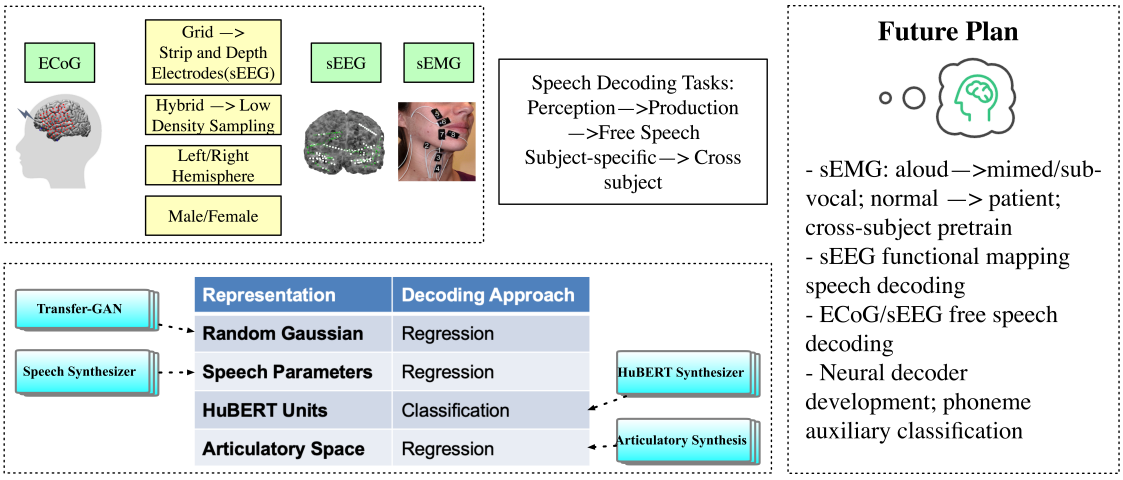


FIGURE 7.1: Conclusion and Future Work

7.1 Conclusion

In this thesis, we presented comprehensive neural speech decoding frameworks leveraging deep learning and speech synthesis techniques, aiming to advance the decoding of neural and muscle activities into speech under conditions of limited data availability. While motivated in part by the goal of restoring communication abilities in individuals with severe speech impairments, our work focuses more broadly on addressing the scientific and technical challenges of

mapping neural or muscular signals to speech representations, aiming to support a wide range of clinical and research applications.

We systematically investigated three neural and muscle signal modalities—ECoG, sEEG, and sEMG. For ECoG, we covered configurations ranging from high-density cortical grids to hybrid or low-density montages, with additional strip and depth electrodes when available. We further explored sEEG as a depth electrode–only modality. Across both ECoG and sEEG, our study included recordings from the left and right hemispheres, and bilateral implantations, encompassing both male and female participants. This comprehensive dataset allowed us to evaluate decoding performance under diverse anatomical, demographic, and recording conditions, providing insights into the generalizability and robustness of our framework.

Across these modalities, we evaluated multiple representation spaces, including random Gaussian embeddings [11], speech parameters [20], HuBERT units [17], and articulatory space [18], coupled with regression or classification-based decoding strategies.

We introduced novel decoder architectures, including the SwinTW transformer for cross-subject generalization and multi-task models with phoneme auxiliary supervision to enhance articulatory feature learning. Through comprehensive evaluations, we demonstrated that our framework enables intelligible and naturalistic speech reconstruction, even under challenging conditions such as low-density recordings and cross-subject generalization. Crucially, this work establishes the feasibility of using sEEG and sEMG for speech decoding, opening the door to less invasive and clinically scalable neural speech prostheses. Taken together, this thesis advances both the scientific understanding and technological development of neural and muscular speech decoding, laying a critical foundation for future translational applications in clinical neuroprosthetics and

human–machine communication.

7.2 Future Work

Building upon the advances reported in this thesis, several promising directions emerge for future research. For sEMG, we plan to extend decoding from aloud to mimed and sub-vocal speech, and to transition from healthy participants to patients, employing cross-subject pretraining to overcome data scarcity. In the sEEG domain, we will explore speech decoding in the word-level functional task and advance toward decoding free speech in both ECoG and sEEG settings.

From a modeling perspective, we aim to develop next-generation neural decoders by incorporating recent advances of deep neural networks [100–102]. We also plan to refine auxiliary tasks such as phoneme classification to serve as inductive biases, enhancing articulatory space learning.

To address the critical challenge of limited patient data, we plan to leverage zero-shot text-to-speech (TTS) systems to generate diverse proxy speech signals paired with limited sEMG or neural recordings. This data augmentation strategy is expected to improve model robustness by encouraging the disentanglement of invariant phonemic content from variable acoustic characteristics. Beyond data augmentation, an important future direction is to pretrain neural representation extractors using contrastive learning [103] or next-token prediction objectives, as in large language models (LLMs [104, 105]) on large-scale neural datasets (for example, our ECoG free speech dataset). Such pretrained encoders can serve as robust backbones for downstream speech decoding tasks, enabling effective transfer learning and reducing the dependence on large amounts of subject-specific data. Combined with domain adaptation techniques [106–108], these models hold promise for rapid adaptation to new patients with minimal

calibration. Furthermore, insights from general few-shot and zero-shot learning paradigms [109–111] can be leveraged to enhance cross-subject generalization, paving the way toward scalable and clinically applicable silent speech interfaces.

Collectively, these directions have the potential to transform neural speech decoding into a scalable, robust, and clinically deployable technology, offering unprecedented communication possibilities for individuals with severe speech impairments.

Bibliography

[1] Tanja Schultz, Michael Wand, Thomas Hueber, Dean J Krusienski, Christian Herff, and Jonathan S Brumberg, "Biosignal-based spoken communication: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2257–2271, 2017.

[2] Kai J Miller, Dora Hermes, and Nathan P Staff, "The current state of electrocorticography-based brain–computer interfaces," *Neurosurgical focus*, vol. 49, no. 1, pp. E2, 2020.

[3] Shiyu Luo, Qinwan Rabbani, and Nathan E Crone, "Brain-computer interface: applications to speech decoding and synthesis to augment communication," *Neurotherapeutics*, vol. 19, no. 1, pp. 263–273, 2022.

[4] David A Moses, Matthew K Leonard, Joseph G Makin, and Edward F Chang, "Real-time decoding of question-and-answer speech dialogue using human cortical activity," *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.

[5] David A Moses, Sean L Metzger, Jessie R Liu, Gopala K Anumanchipalli, Joseph G Makin, Pengfei F Sun, Josh Chartier, Maximilian E Dougherty, Patricia M Liu, Gary M Abrams, Adelyn Tu-Chan D.O., Karunesh Ganguly, and Edward F. Chang, "Neuroprosthesis for decoding speech in a paralyzed person with anarthria," *New England Journal of Medicine*, vol. 385, no. 3, pp. 217–227, 2021.

[6] Christian Herff and Tanja Schultz, "Automatic speech recognition from neural signals: a focused review," *Frontiers in neuroscience*, vol. 10, pp. 429, 2016.

[7] Qinwan Rabbani, Griffin Milsap, and Nathan E Crone, "The potential for a speech brain–computer interface using chronic electrocorticography," *Neurotherapeutics*, vol. 16, no. 1, pp. 144–165, 2019.

[8] Miguel Angrick, Christian Herff, Emily Mugler, Matthew C Tate, Marc W Slutzky, Dean J Krusienski, and Tanja Schultz, "Speech synthesis from ecog using densely connected 3d convolutional neural networks," *Journal of neural engineering*, vol. 16, no. 3, pp. 036019, 2019.

[9] Pengfei Sun, Gopala K Anumanchipalli, and Edward F Chang, "Brain2char: a deep architecture for decoding text from brain recordings," *Journal of neural engineering*, vol. 17, no. 6, pp. 066015, 2020.

[10] Joseph G Makin, David A Moses, and Edward F Chang, "Machine translation of cortical activity to text with an encoder–decoder framework," *Nature neuroscience*, vol. 23, no. 4, pp. 575–582, 2020.

[11] Ran Wang, Xupeng Chen, Amirhossein Khalilian-Gourtani, Zhaoxi Chen, Leyao Yu, Adeen Flinker, and Yao Wang, "Stimulus speech decoding from human cortex with generative adversarial network transfer learning," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 390–394.

[12] Petr Zelinka, Milan Sigmund, and Jiri Schimmel, "Impact of vocal effort variability on automatic speech recognition," *Speech Communication*, vol. 54, no. 6, pp. 732–742, 2012.

[13] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al., "Automatic speech recognition and speech variability: A review," *Speech communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[14] Sean L Metzger, Kaylo T Littlejohn, Alexander B Silva, David A Moses, Margaret P Seaton, Ran Wang, Maximilian E Dougherty, Jessie R Liu, Peter Wu, Michael A Berger, et al., "A high-performance neuroprosthesis for speech decoding and avatar control," *Nature*, pp. 1–10, 2023.

[15] Miguel Angrick, Maarten Ottenhoff, Lorenz Diener, Darius Ivucic, Gabriel Ivucic, Sophocles Goulis, Albert J. Colon, Louis Wagner, Dean J. Krusienski, Pieter L. Kubben, Tanja Schultz, and Christian Herff, "Towards closed-loop speech synthesis from stereotactic eeg: A unit selection approach," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1296–1300.

[16] Jonas Kohler, Maarten C Ottenhoff, Sophocles Goulis, Miguel Angrick, Albert J Colon, Louis Wagner, Simon Tousseyn, Pieter L Kubben, and Christian Herff, "Synthesizing speech from intracranial depth electrodes using an encoder-decoder framework," *arXiv preprint arXiv:2111.01457*, 2021.

[17] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[18] Cheol Jun Cho, Peter Wu, Tejas S Prabhune, Dhruv Agarwal, and Gopala K Anumanchipalli, "Coding speech through vocal tract kinematics," *IEEE Journal of Selected Topics in Signal Processing*, 2024.

[19] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, "DDSP: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[20] Xupeng Chen, Ran Wang, Amirhossein Khalilian-Gourtani, Leyao Yu, Patricia Dugan, Daniel Friedman, Werner Doyle, Orrin Devinsky, Yao Wang, and Adeen Flinker, "A neural speech decoding framework leveraging deep learning and speech synthesis," *Nature Machine Intelligence*, pp. 1–14, 2024.

[21] Junbo Chen, Xupeng Chen, Ran Wang, Chenqian Le, Amirhossein Khalilian-Gourtani, Erika Jensen, Patricia Dugan, Werner Doyle, Orrin Devinsky, Daniel Friedman, et al., "Transformer-based neural speech decoding from surface and depth electrode signals," *Journal of Neural Engineering*, vol. 22, no. 1, pp. 016017, 2025.

[22] Stéphanie Martin, Peter Brunner, Chris Holdgraf, Hans-Jochen Heinze, Nathan E Crone, Jochem Rieger, Gerwin Schalk, Robert T Knight, and Brian N Pasley, "Decoding spectrotemporal features of overt and covert speech from the human cortex," *Frontiers in neuroengineering*, vol. 7, pp. 14, 2014.

[23] Christian Herff, Garett Johnson, Lorenz Diener, Jerry Shih, Dean Krusienski, and Tanja Schultz, "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)*. IEEE, 2016, pp. 1540–1543.

[24] M Angrick, MC Ottenhoff, L Diener, D Ivucic, G Ivucic, S Goulis, J Saal, AJ Colon, L Wagner, DJ Krusienski, et al., "Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity. commun biol 4 (1): 1055–1055," 2021.

[25] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang, "Speech synthesis from neural decoding of spoken sentences," *Nature*, vol. 568, no. 7753, pp. 493–498, 2019.

[26] Christian Herff, Lorenz Diener, Miguel Angrick, Emily Mugler, Matthew C Tate, Matthew A Goldrick, Dean J Krusienski, Marc W Slutzky, and Tanja Schultz, "Generating natural, intelligible speech from brain activity in motor, premotor, and inferior frontal cortices," *Frontiers in neuroscience*, vol. 13, pp. 1267, 2019.

[27] Xiaolong Wu, Scott Wellington, Zhichun Fu, and Dingguo Zhang, "Speech decoding from stereo-electroencephalography (seeg) signals using advanced deep learning methods," *Journal of Neural Engineering*, 2024.

[28] Simone Graetzer and Carl Hopkins, "Intelligibility prediction for speech mixed with white gaussian noise at low signal-to-noise ratios," *The Journal of the Acoustical Society of America*, vol. 149, no. 2, pp. 1346–1362, 2021.

[29] John Kominek, Tanja Schultz, and Alan W Black, "Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion," in *Spoken Languages Technologies for Under-Resourced Languages*, 2008.

[30] Hugging Face, "facebook/hubert-large-ls960-ft," `https://huggingface.co/facebook/hubert-large-ls960-ft`, 2024, Accessed: 2025-04-21.

[31] Vladimir I Levenshtein et al., "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*. Soviet Union, 1966, vol. 10, pp. 707–710.

[32] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[33] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2010, pp. 4214–4217.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.

[36] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] Edward F Chang, Kunal P Raygor, and Mitchel S Berger, "Contemporary model of language organization: an overview for neurosurgeons," *Journal of neurosurgery*, vol. 122, no. 2, pp. 250–261, 2015.

[38] Jintao Jiang, Marcia Chen, and Abeer Alwan, "On the perception of voicing in syllable-initial plosives in noise," *The Journal of the Acoustical Society of America*, vol. 119, no. 2, pp. 1092–1105, 2006.

[39] Ran Wang, Xupeng Chen, Amirhossein Khalilian-Gourtani, Leyao Yu, Patricia Dugan, Daniel Friedman, Werner Doyle, Orrin Devinsky, Yao Wang, and Adeen Flinker, "Distributed feedforward and feedback cortical processing supports human speech production," *Proceedings of the National Academy of Sciences*, vol. 120, no. 42, pp. e2300255120, 2023.

[40] James L Flanagan, "A difference limen for vowel formant frequency," *The journal of the Acoustical Society of America*, vol. 27, no. 3, pp. 613–617, 1955.

[41] Ronald W Schafer and Lawrence R Rabiner, "System for automatic formant analysis of voiced speech," *The Journal of the Acoustical Society of America*, vol. 47, no. 2B, pp. 634–648, 1970.

[42] James L Fitch and Anthony Holbrook, "Modal vocal fundamental frequency of young adults," *Archives of Otolaryngology*, vol. 92, no. 4, pp. 379–382, 1970.

[43] Stanley S Stevens and John Volkmann, "The relation of pitch to frequency: A revised scale," *The American Journal of Psychology*, vol. 53, no. 3, pp. 329–353, 1940.

[44] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] Paul Boersma and Vincent Van Heuven, "Speak and unspeak with praat," *Glot International*, vol. 5, no. 9/10, pp. 341–347, 2001.

[46] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[47] Jennifer Shum, Lora Fanda, Patricia Dugan, Werner K Doyle, Orrin Devinsky, and Adeen Flinker, "Neural correlates of sign language production revealed by electrocorticography," *Neurology*, vol. 95, no. 21, pp. e2880–e2889, 2020.

[48] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner, "Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires," *PLoS computational biology*, vol. 16, no. 10, pp. e1008228, 2020.

[49] Gregory Hickok and David Poeppel, "The cortical organization of speech processing," *Nature Reviews Neuroscience*, vol. 8, no. 5, pp. 393, 2007.

[50] Lydia A Trupe, Daniel D Varma, Yessenia Gomez, David Race, Richard Leigh, Argye E Hillis, and Rebecca F Gottesman, "Chronic apraxia of speech and broca's area," *Stroke*, vol. 44, no. 3, pp. 740–744, 2013.

[51] Emily M Mugler, Matthew C Tate, Karen Livescu, Jessica W Templer, Matthew A Goldrick, and Marc W Slutzky, "Differential representation of articulatory gestures and phonemes in precentral and inferior frontal gyri," *Journal of Neuroscience*, pp. 1206–18, 2018.

[52] Christian Herff, Dominic Heger, Adriana De Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz, "Brain-to-text: decoding spoken phrases from phone representations in the brain," *Frontiers in neuroscience*, vol. 9, pp. 217, 2015.

[53] Muge Ozker, Werner Doyle, Orrin Devinsky, and Adeen Flinker, "A cortical network processes auditory error signals during human speech production to maintain fluency," *PLoS biology*, vol. 20, no. 2, pp. e3001493, 2022.

[54] Andrew Stuart, Joseph Kalinowski, Michael P Rastatter, and Kerry Lynch, "Effect of delayed auditory feedback on normal speakers at two speech rates," *The Journal of the Acoustical Society of America*, vol. 111, no. 5, pp. 2237–2241, 2002.

[55] Maxime Verwoert, Maarten C Ottenhoff, Sophocles Goulis, Albert J Colon, Louis Wagner, Simon Tousseyn, Johannes P van Dijk, Pieter L Kubben, and Christian Herff, "Dataset of speech production in intracranial electroencephalography," *Scientific data*, vol. 9, no. 1, pp. 434, 2022.

[56] Julia Berezutskaya, Zachary V Freudenburg, Mariska J Vansteensel, Erik J Aarnoutse, Nick F Ramsey, and Marcel AJ van Gerven, "Direct speech reconstruction from sensorimotor brain activity with optimized deep learning models," *Journal of Neural Engineering*, vol. 20, no. 5, pp. 056010, 2023.

[57] Ran Wang, Yao Wang, and Adeen Flinker, "Reconstructing speech stimuli from human auditory cortex activity using a WaveNet approach," in *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE, 2018, pp. 1–6.

[58] James L Flanagan, *Speech analysis synthesis and perception*, vol. 3, Springer Science & Business Media, 2013.

[59] Xavier Serra and Julius Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[60] Gregory B Cogan, Thomas Thesen, Chad Carlson, Werner Doyle, Orrin Devinsky, and Bijan Pesaran, "Sensory–motor transformations for speech occur bilaterally," *Nature*, vol. 507, no. 7490, pp. 94–98, 2014.

[61] Kenji Ibayashi, Naoto Kunii, Takeshi Matsuo, Yohei Ishishita, Seijiro Shimada, Kensuke Kawai, and Nobuhito Saito, "Decoding speech with integrated hybrid signals recorded from the human ventral motor cortex," *Frontiers in neuroscience*, vol. 12, pp. 221, 2018.

[62] Pedram Z Soroush, Christian Herff, Stephanie K Ries, Jerry J Shih, Tanja Schultz, and Dean J Krusienski, "The nested hierarchy of overt, mouthed, and imagined speech activity evident in intracranial recordings," *NeuroImage*, vol. 269, pp. 119913, 2023.

[63] Matthew C Tate, Guillaume Herbet, Sylvie Moritz-Gasser, Joseph E Tate, and Hugues Duffau, "Probabilistic map of critical functional regions of the human cerebral cortex: Broca's area revisited," *Brain*, vol. 137, no. 10, pp. 2773–2782, 2014.

[64] Michael A Long, Kalman A Katlowitz, Mario A Svirsky, Rachel C Clary, Tara McAllister Byun, Najib Majaj, Hiroyuki Oya, Matthew A Howard, and Jeremy DW Greenlee, "Functional segregation of cortical regions underlying speech timing and articulation," *Neuron*, vol. 89, no. 6, pp. 1187–1193, 2016.

[65] Francis R Willett, Erin M Kunz, Chaofei Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al., "A high-performance speech neuroprosthesis," *Nature*, pp. 1–6, 2023.

[66] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211.

[67] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[68] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al., "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12009–12019.

[69] Shuji Komeiji, Kai Shigemi, Takumi Mitsuhashi, Yasushi Iimura, Hiroharu Suzuki, Hidenori Sugano, Koichi Shinoda, and Toshihisa Tanaka, "Transformer-based estimation of spoken sentences using electrocorticography," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1311–1315.

[70] Kai Shigemi, Shuji Komeiji, Takumi Mitsuhashi, Yasushi Iimura, Hiroharu Suzuki, Hidenori Sugano, Koichi Shinoda, Kohei Yatabe, and Toshihisa Tanaka, "Synthesizing speech from ecog with a combination of transformer-based encoder and neural vocoder," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[71] Koji Iida and Hiroshi Otsubo, "Stereoelectroencephalography: indication and efficacy," *Neurologia medico-chirurgica*, vol. 57, no. 8, pp. 375–385, 2017.

[72] Nitin Tandon, Brian A Tong, Elliott R Friedman, Jessica A Johnson, Gretchen Von Allmen, Melissa S Thomas, Omotola A Hope, Giridhar P Kalamangalam, Jeremy D Slater, and Stephen A Thompson, "Analysis of morbidity and outcomes associated with use of subdural grids vs stereoelectroencephalography in patients with intractable epilepsy," *JAMA neurology*, vol. 76, no. 6, pp. 672–681, 2019.

[73] Christian Herff, Dean J Krusienski, and Pieter Kubben, "The potential of stereotactic-eeg for brain-computer interfaces: current progress and future directions," *Frontiers in neuroscience*, vol. 14, pp. 123, 2020.

[74] Jyun Senda, Mai Tanaka, Keiya Iijima, Masato Sugino, Fumina Mori, Yasuhiko Jimbo, Masaki Iwasaki, and Kiyoshi Kotani, "Auditory stimulus reconstruction from ecog with dnn and self-attention modules," *Biomedical Signal Processing and Control*, vol. 89, pp. 105761, 2024.

[75] Alexandre Défossez, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King, "Decoding speech perception from non-invasive brain recordings," *Nature Machine Intelligence*, vol. 5, no. 10, pp. 1097–1107, 2023.

[76] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[77] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[78] Ronald W Schafer, "What is a savitzky-golay filter?[lecture notes]," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[79] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale

weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28492–28518.

[80] Amirhossein Khalilian-Gourtani, Ran Wang, Xupeng Chen, Leyao Yu, Patricia Dugan, Daniel Friedman, Werner Doyle, Orrin Devinsky, Yao Wang, and Adeen Flinker, "A corollary discharge circuit in human speech (under review)," *Nature Human Behaviour*, pp. 2022–09, 2022.

[81] Miguel Angrick, Maarten Ottenhoff, Lorenz Diener, Darius Ivucic, Gabriel Ivucic, Sophocles Goulis, Albert J Colon, Louis Wagner, Dean J Krusienski, Pieter L Kubben, et al., "Towards closed-loop speech synthesis from stereotactic eeg: a unit selection approach," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1296–1300.

[82] Jonas Kohler, Maarten C Ottenhoff, Sophocles Goulis, Miguel Angrick, Albert J Colon, Louis Wagner, Simon Tousseyn, Pieter L Kubben, and Christian Herff, "Synthesizing speech from intracranial depth electrodes using an encoder-decoder framework," *arXiv preprint arXiv:2111.01457*, 2021.

[83] Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al., "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[84] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel

spectrogram predictions. corr abs/1712.05884 (2017)," *URL: http://arxiv.org/abs/1712.05884*, 2017.

[85] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[86] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[87] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[88] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[89] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[90] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 161–165.

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

[92] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[93] Peter J Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

[94] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[95] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[96] Li Yujian and Liu Bo, "A normalized levenshtein distance metric," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[97] Yuanning Li, Gopala K Anumanchipalli, Abdelrahman Mohamed, Peili Chen, Laurel H Carney, Junfeng Lu, Jinsong Wu, and Edward F Chang, "Dissecting neural computations in the human auditory pathway using deep neural networks for speech," *Nature Neuroscience*, vol. 26, no. 12, pp. 2213–2225, 2023.

[98] Cheol Jun Cho, Nicholas Lee, Akshat Gupta, Dhruv Agarwal, Ethan Chen, Alan W Black, and Gopala K Anumanchipalli, "Sylber: Syllabic embedding representation of speech from raw audio," *arXiv preprint arXiv:2410.07168*, 2024.

[99] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al., "E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts," in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 682–689.

[100] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[101] Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville, "Forgetting transformer: Softmax attention with a forget gate," *arXiv preprint arXiv:2503.02130*, 2025.

[102] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira, "Perceiver: General perception with iterative attention," in *International conference on machine learning*. PMLR, 2021, pp. 4651–4664.

[103] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[104] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[105] Tongtian Yue, Shuning Xue, Xuange Gao, Yepeng Tang, Longteng Guo, Jie Jiang, and Jing Liu, "Eegpt: Unleashing the potential of eeg generalist foundation model by autoregressive pre-training," *arXiv preprint arXiv:2410.19779*, 2024.

[106] Wenhui Guo, Guixun Xu, and Yanjiang Wang, "Multi-source domain adaptation with spatio-temporal feature extractor for eeg emotion recognition," *Biomedical Signal Processing and Control*, vol. 84, pp. 104998, 2023.

[107] Yi-Ming Jin, Yu-Dong Luo, Wei-Long Zheng, and Bao-Liang Lu, "Eeg-based emotion recognition using domain adaptation network," in *2017 international conference on orange technologies (ICOT)*. IEEE, 2017, pp. 222–225.

[108] Rushuang Zhou, Weishan Ye, Zhiguo Zhang, Yanyang Luo, Li Zhang, Linling Li, Gan Huang, Yining Dong, Yuan-Ting Zhang, and Zhen Liang, "Eegmatch: Learning with incomplete labels for semisupervised eeg-based cross-subject emotion recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[109] Sion An, Soopil Kim, Philip Chikontwe, and Sang Hyun Park, "Few-shot relation learning with attention for eeg-based motor imagery classification," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10933–10938.

[110] Sachin Ravi and Hugo Larochelle, "Optimization as a model for few-shot learning," in *International conference on learning representations*, 2017.

[111] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel, "Meta-learning for semi-supervised few-shot classification," *arXiv preprint arXiv:1803.00676*, 2018.

List of Publications

1. Chen, J., Chen, X., Wang, R., Le, C., Khalilian-Gourtani, A., Jensen, E., Dugan, P., Doyle, W., Devinsky, O., Friedman, D., Flinker, A and Wang, Y. "Transformer-based neural speech decoding from surface and depth electrode signals" Journal of Neural Engineering, 2025.

2. Chen, X., Wang, R., Khalilian-Gourtani, A., Yu, L., Dugan, P., Friedman, D., Doyle, W., Devinsky, O., Wang, Y. and Flinker, A. "A Neural Speech Decoding Framework Leveraging Deep Learning and Speech Synthesis." Nature Machine Intelligence, 2024.

3. Khalilian-Gourtani, A., Wang, R., Chen, X., Yu, L., Dugan, P., Friedman, D., Doyle, W., Devinsky, O., Wang, Y. and Flinker, A. "A corollary discharge circuit in human speech." Proceedings of the National Academy of Sciences (PNAS), 2024.

4. Wang, R., Chen, X., Khalilian-Gourtani, A., Yu, L., Dugan, P., Friedman, D., Doyle, W., Devinsky, O., Wang, Y. and Flinker, A. "Distributed feedforward and feedback cortical processing supports human speech production." Proceedings of the National Academy of Sciences (PNAS), 2023.

5. Wang, R., Chen, X., Khalilian-Gourtani, A., Chen, Z., Yu, L., Flinker, A. and Wang, Y. "Stimulus speech decoding from human cortex with generative adversarial network transfer learning." IEEE 17th International Symposium on Biomedical Imaging (ISBI), 2020.