2024

Adaptable and Trustworthy Machine Learning for Human Activity Recognition from Bioelectric Signals

Morgan S. Stuart
Virginia Commonwealth University

Adaptable and Trustworthy Machine Learning for Human Activity Recognition from Bioelectric Signals

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University

by

Morgan Simmons Stuart

Director: Milos Manic

Professor, Department of Computer Science

Virginia Commonwealth University

Richmond, Virginia

April 2024

Acknowledgements

To my mother Bess, thank you for teaching me to enjoy the world outside of computers and always encouraging me to pursue my interests. To my father Andy, thank you for showing me how to persevere and inspiring me to pursue engineering. To my brother Drew, thank you for "developing" the fight in me - it's surprisingly useful in research. To Annie, thank you for supporting me every step of the way, day-in and day-out.

I have been fortunate to have collaborated with a small tribe of fantastic people while completing this work. To my lab peers in the Modern Heuristics Research Group - Victor, Sandun, Daniel, Chathurika, Kasun, and Jake - thank you for the earnest and thoughtful discussions. Across labs, I would like to extend my thanks to Deepak and Dr. Holloway for entrusting me with your research questions. From that work, I'm grateful to have been connected with Dr. Krusienski and Srdjan. Srdjan, somehow you manage to be both grandiose and humble, with a clear passion for improving the lives of those lucky enough to enjoy the space around you. However small, I think we left our mark on the history of human-computer interfaces.

To friends and colleagues, thank you for your patience while I spent time completing this dissertation. I'd like to especially thank Matt, both for showing me the ropes of data science in industry and simply giving me the space to start this work. I'd also like to thank Paul, who without a degree, is still one of the best engineers I've ever met, thank you for sharing your work with me.

I would also like to thank my advisor Prof. Milos Manic for his patience and guidance - it was your course during my masters that finalized my desire to embark on my doctorate. I would also like to sincerely thank Dr. Dean Krusienski, Dr. Kathryn Holloway, Dr. Alberto Cano, and Dr. Cang Ye for serving on my committee and providing helpful feedback.

TABLE OF CONTENTS

ABSTRACT

Enabling machines to learn measures of human activity from bioelectric signals has many applications in human-machine interaction and healthcare. However, labeled activity recognition datasets are costly to collect and highly varied, which challenges machine learning techniques that rely on large datasets. Furthermore, activity recognition in practice needs to account for user trust - models are motivated to enable interpretability, usability, and information privacy.

The objective of this dissertation is to improve adaptability and trustworthiness of machine learning models for human activity recognition from bioelectric signals. We improve adaptability by developing pretraining techniques that initialize models for later specialization to unseen users. We further expand adaptability with models that pretrain from unlabeled data, and can support transfer learning to both new users and new tasks. We address the need for improved trust with an engineering-informed approach to integrated explainability and reduced model complexity. We also investigate training dataset privacy, another component of trustworthy model building, with methods to evaluate information leakage in our neural representation extraction models.

The intersection of adaptability and trustworthiness is especially critical for human activity recognition from bioelectric signals. Modeling human physiology has broad applications across many domains and, due to its highly personalized nature, must meet high expectations of trust. These are met in part when users can maintain privacy and understand how the system using their data operates. Further heightening the need for trustworthy systems is the variability in bioelectric sensor data, which can often uniquely identify different users and their system configurations apart from others. It is this same variability that makes generalized machine learning models for human activity recognition from bioelectric signals challenging. Aspects like varying electrode locations, skin salinity, power-line noise, all conflate with the wide diversity of human behavior and physiology to make adaptability difficult.

In order to address these challenges, we present an interpretable convolutional model for the task of speech detection from neural signals measured with electrocorticography, and we validate that it discovered features consistent with current neuroscience literature. We use a transfer learning approach to adapt hand-pose recognition models to new users with commodity electromyography devices, without the need for expensive pre-processing used in prior work. We develop an approach for self-supervised learning from stereotactic electoencephalogram signals, enabling adaptation to unseen tasks and users from unlabeled data. Finally, in order to understand privacy risks, we assess information leakage using re-identification and membership inference attacks against our neural representation learning methodology.

Our work, as applied to existing and emerging human-computer interfaces, demonstrates that machine learning can be made to both support human well-being and adapt to our complexity, without abandoning pursuit of user trust.

CHAPTER 1

INTRODUCTION

The continuous delivery of bioelectric sensor readings enables low-latency monitoring and decision-making in healthcare, industry, and every-day life. Compared with other modalities such as video, bioelectric sensor signals are often a cost-effective and more personalized method to recognize activities, intent, or even treatment response.

However, information-rich streams of bioelectric data can be challenging to use in practice. Training data of labeled human activities, aligned with their bioelectric signals, are expensive to collect and are therefore often small. These smaller datasets make it difficult to develop general-purpose models without over-fitting. Models are further challenged with feature drift from new participants, evolving activities, and possibly modified sensor configurations. The resulting distribution discrepancies across small datasets motivate focus on *adaptable* models that account for these issues - models capable of adapting to new participants, new classification tasks, and new sensor positions.

While adaptability is important for predictive performance, models must also cultivate trust by supporting privacy-conscious use-cases, addressing information leakage, and implementing interpretable processes. Trustworthy systems are also *usable*, meaning that they support human autonomy, with locally executable solutions that do not need to transmit sensitive data to third-party systems. To aid in validation, troubleshooting, and improved trust, methods should strive to be interpretable. When providing personal data to a database for model training, users risk information leakage and possible re-identification, which should be both measured and mitigated to improve trust.

In this dissertation, our objective is to improve adaptability and trust of human activity recognition models. We improve adaptability with transfer learning and self-supervised learning. Transfer learning takes advantage of prior source data to better adapt to new target data, while self-supervised learning leverages unlabeled data to establish an adaptable model.

To improve trust, this work's experiments address interpretrability, usability, and privacy. We construct engineering-informed deep neural networks for a time-series data modality to enable interpretable models that are more usable on personal devices. The resulting model provides insights into what parts of the frequency spectrum it utilizes for its task, while greatly reducing model complexity for potential deployment on resource-constrained systems. We also research the privacy risks of our method for learning neural representations using novel patient-centered information leakage experiments. In Figure 1, we illustrate the model life-cycle for our domain, and the objectives our methods will address in this dissertation.

1.1 Motivations

Machine Learning (ML) systems [1, 2, 3] interacting with human physiology are faced with sensitive and highly variable inputs [4, 5, 6]. Different *users* perform tasks in ways that suit their physiology, and supporting hardware systems vary in their capabilities. These realities of the implicit variance in such systems create distribution *discrepancies* that leak information and confound common data-driven learning approaches. Successful methods must adapt to these changes, often optimizing models per unique user and configuration. However, per-user models require extensive data collection and training time before a user can use their model. These methods avoid the issue of generalizing information from many individuals and their respective configurations.

We frame the Human Activity Recognition (HAR) modeling workflow and highlight the areas our work contributes in Figure 1. While many HAR modalities exist [7, 8], this dissertation targets bioelectric sensor data and its unique challenges [5]. In this dissertation, we assume a *provider* builds or otherwise prepares a model that can be deployed to a user's local system. Models are prepared by generalizing information from a training set population, but may again undergo further specialization once deployed. Our adaptability goals refer to both successful specialization (i.e., adaptation *to a new domain*) as well as successfully generalizing from broader dataset definitions (i.e., adaptation *from several domains*) [9].

In order to broaden application, adoption, and ethical use, providers of human activity recognition also must pursue *trustworthy* models that are interpretable and are usable on commodity hardware for local and private computation. As adoption grows, so will large stores of labeled and unlabeled data. Increased coverage and dataset robustness will improve models, but these models may still leak user information. Before release, providers of ML solutions should closely inspect their models and may consider steps to mitigate risks of implicit information leakage [10]. This is especially important for more trustworthy deployments, in which a user is provided control over their model, since an adversarial user may use this autonomy to extract information about the training dataset.

1.1.1 Adaptable Generalization and Specialization

ML systems must generalize from a training procedure to their intended application - this adaptation is foundational [1, 2]. In ideal scenarios, generalization is obtained with common modeling approaches and simple regularization to prevent over-fitting. However, for systems that commonly face underlying distribution discrepancy, model construction is motivated to consider generalization more closely [11]. With more data, larger models can be built that capture more complexity, but when data size is reduced, discrepancies make generalization difficult. In these scenarios, applying a more significant prior through informed model designs can help make better use of both smaller datasets and larger datasets alike [12]. From larger, possibly unlabeled datasets, models must learn general forms of the data domain, ones that are able to translate to downstream tasks of interest. This is common in the domain of natural language processing [13, 14]. Even with robust generalization, some amount of specialization to a new user or perhaps configuration can greatly improve predictive performance. However, this user-specific specialization procedure should occur locally to improve privacy and trust [15]. Local systems have additional restrictions since they are often resource constrained platforms [16]. The importance of autonomy and privacy are discussed in our background in Chapter 2, Subsections 2.1.2 and 2.1.3. We discuss methods for ML trust and adaptability as part of Subsection 2.4.

1.1.2 Trustworthy Contribution and Use

Intelligent recognition systems for human activity must reduce risks and improve adoption through trustworthy approaches [15]. Trust has many facets, but our work is focused on interpretability[17, 18] , privacy [19, 19, 20, 21], and usability [15, 22, 23] in Artificial Intelligence (AI). More trustworthy systems enable insight into what information drives prediction. Interpretable systems also allow expert to validate findings from previous work or make novel discoveries. As illustrated in Figure 1, the need for interpretability exists both when generalizing from large cohorts and specializing to specific tasks. Both scenarios can improve trust through explainable processes. Information-rich sensor data, collected over time, provides significant detail on the individual. Of course, without this detail, data-driven solutions for activity recognition may not be viable. Given this sensitive data, models and systems can more easily maintain trust by simply not requiring that the data leave the user's own systems. Practically, this requires models that can be adapted to the new user and deployed, all within a trusted consumer-grade edge system. Any system that requires a database of many users' data, or a large enterprise system, will undermine user trust [19] and require resources for data transmission. Still, the entanglement of sensitive attributes with causal signals may be discernible from the model itself [24]. An adversary may rightfully request access to a model, then use this access to infer an individuals membership in a dataset [25, 26, 27]. For instance, an adversary may be seek to determine if an individual has a particular disease by determining if their data was used in the dataset to train a model released for treatment evaluation. Existing research and policy development on trustworthy AI is further reviewed in Section 2.1.

1.1.3 Application: Human Activity Recognition from Bioelectric Signals

The primary goal of human activity recognition is to provide actionable insights by monitoring sensor data continuously. In pursuit of improved performance, popular Deep Learning (DL) [28] architectures are applied to sensor data for activity recognition [29]. Architectures such as a Convolutional Neural Network (CNN) [30, 31, 32], Recurrent Neural Network (RNN) [33], or a combination [34, 35, 36] have been used to classify activities from sensors. These models achieve state-of-the-art results, but often require upwards of hundreds of thousands of learnable parameters. To reduce processing overhead, some efforts use general-purpose model reduction techniques [37] or avoid raw data entirely[38]. Furthermore, DL classification models borrowed from other domains, such as CNNs designed for image classification, do not leverage the unique characteristics of sensor data for HAR. Sensor data evenly sampled over time yields a time-series dataset, a data modality whose covariance and assumptions diverge from image data. We introduce HAR through the lens of Human Computer Interaction (HCI) and discuss the domain's challenges further in Section 2.2. General DL methods which we apply throughout this work are outlined in Section 2.3, with additional discussion of methodologies relevant to trust and adaption in Section 2.4.

1.2 Goals and Contributions

Objective: Improve adaptability and trust of machine learning for human activity recognition systems.

1. **Goal:** Improve interpretability and reduce complexity by learning engineering-informed models.
   Contribution: SincIEEG uses Sinc-Net to improve performance and provide insights for neural speech decoding

   - Reduce complexity with engineering-informed model design
   - Enable interpretability with engineering-informed model design
   - Validate discovery of task-relevant spectral features captured directly from data by an interpretable model

2. **Goal:** Improve inter-person adaptation using transfer learning across individuals
   Contribution: SincEMG uses transfer learning and Sinc-Net layers to reduce parameters and improve performance

   - Leverage transfer learning to reduce model size and improve performance for activity recognition on raw sensor data.
   - Visualize models adapting to unseen users during fine-tuning
   - Achieve 10x reduction in number of learned parameters with comparable or superior accuracy

3. **Goal:** Enable adaptation from unlabeled data with self-supervised pretraining
   Contribution: Brain2Vec architecture for self-supervised pretraining from neural data, applied to neural speech decoding.

   - self-supervised learning of neural representations from unlabeled data
   - Demonstrate learned representations with fine-tuning classification tasks of a neural signals using pretrained Brain2Vec models

4. **Goal:** Characterize neural representations and their privacy risks to individuals
   Contribution: Grid search of brain2vec across varied pretraining cohorts and two simulated privacy attack experiments.

   - Hyperparameter changes vary pretraining performance, but have less impact on fine-tuning performance
   - Neural representations produced by brain2vec can support person re-identification
   - Shadow modeling membership inference attacks appear feasible, but inconsistent

1.3 Organization of this Dissertation

The remainder of this dissertation is organized as follows:

- **Chapter 2** provides background on the concepts of trustworthiness, human activity recognition, and machine learning. Includes work from [16].
- **Chapter 3** introduces Multi-SincNet, an explainable method of machine learning applied to bioelectric neural signals for speech detection. Includes work from [39] [40].
- **Chapter 4** presents a transfer learning method that uses Multi-SincNet to pretrain from many users commodity electromyographic signal data. Includes work from [41]
- **Chapter 5** introduces brain2vec, a methodology that learns from unlabeled neural signals to later adapt to new users and new classification tasks with labeled data. We perform an extended hyperparameter search across different participant cohort sizes and perform experiments to assess the risk of inforamtion leakage. Includes work from [42].
- **Chaper 6** concludes this work with discussion of recent developments and direction for future efforts.

Collaboration Background: The ideation, design, and implementation of the work supporting Section 3.3 [40] & Section 5.2 [42] were completed in close collaboration with then PhD candidate Srdjan Lesaja from the department of biomedical engineering. Each of these two supporting works are published with equal contribution from both myself and Srdjan Lesaja.

Our research resulting in SincIEEG (ch. 3, sec. 3.3) stemmed from discussion about the intersection of our interests in adaptable yet parsimonious solutions to human-machine interfacing. My claims in this work are the interpretability and reduced complexity of the resulting engineering-informed model. Srdjan claims reduced pre-processing and per-patient adaptability of what are typically expert-derived features.

Our continuing research has resulted in brain2vec (ch. 5, sec. 5.2) as a method to leverage more recent self-supervised pretraining techniques for sequential data. My claims in Section 5.2 are the adaptability from unlabeled data using self-supervised learning. Srdjan claims the encoding of spatial information and biologically inspired hidden-unit encoding.

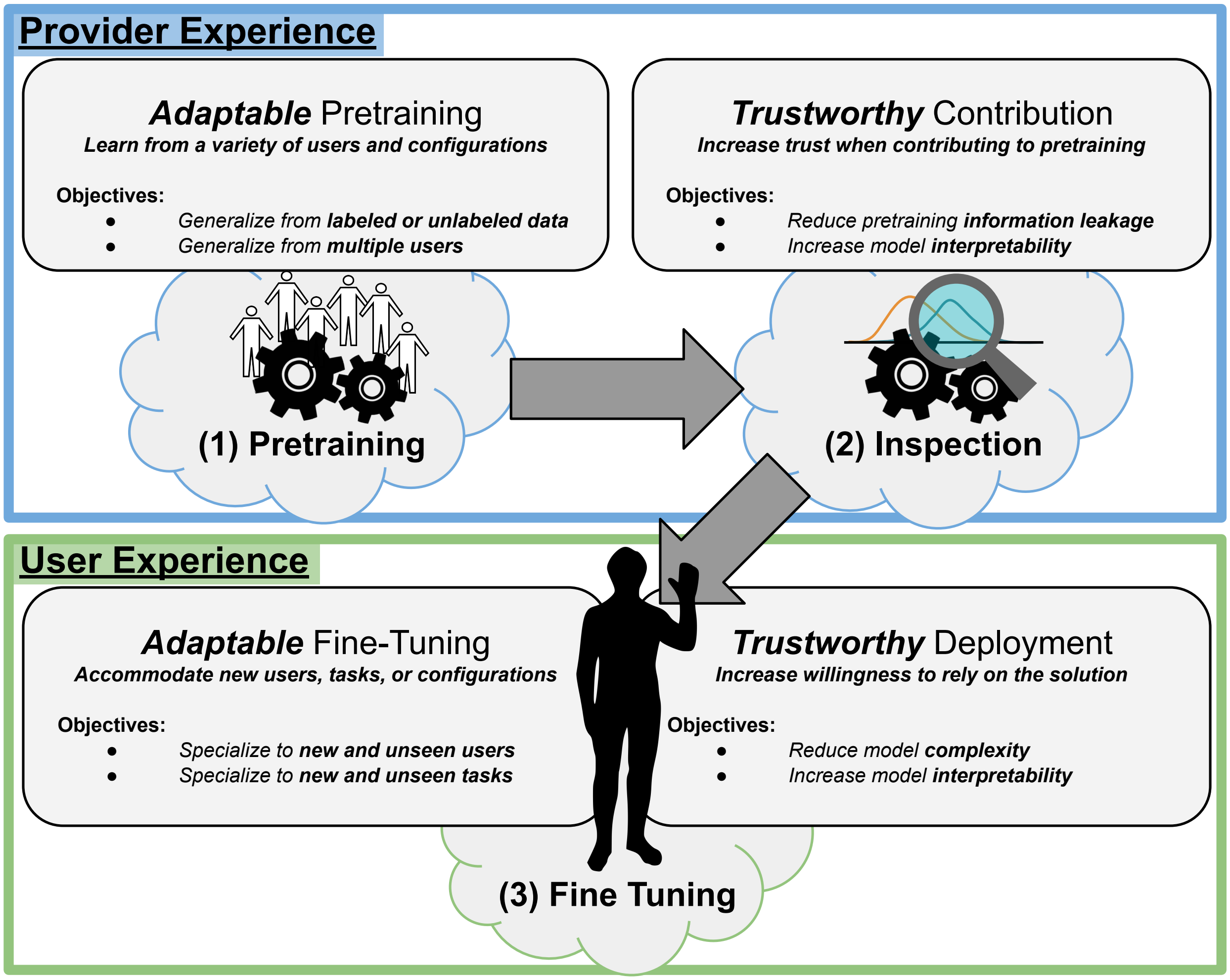

Provider Experience
Adaptable Pretraining
Learn from a variety of users and configurations
Objectives:
• Generalize from labeled or unlabeled data
• Generalize from multiple users
Trustworthy Contribution
Increase trust when contributing to pretraining
Objectives:
• Reduce pretraining information leakage
• Increase model interpretability
(1) Pretraining
(2) Inspection
User Experience
Adaptable Fine-Tuning
Accommodate new users, tasks, or configurations
Objectives:
• Specialize to new and unseen users
• Specialize to new and unseen tasks
Trustworthy Deployment
Increase willingness to rely on the solution
Objectives:
• Reduce model complexity
• Increase model interpretability
(3) Fine Tuning

Fig. 1. Workflow illustration that represents our approach to providing human activity recognition, highlighting our objectives in each area. First, in step **(1)**, a provider of a activity recognition model pretrains their model on collected data. Ideally, this phase is capable of utilizing labeled or unlabeled data, and is capable of learning from multiple users and their configurations. The intention is to capture more information in pretraining phase, such that fine tuning for a new user or configuration can happen more quickly or with less data. Next, in step **(2)** users and experts are made aware of what the model has learned and any unexpected biases. Steps may be taken to debias the model in order to avoid information leakage to adversarial end-users. Steps **(1)** and **(2)** may happen together in practice, for instance, when a model is regularized during pretraining to avoid learning unwanted biases. Finally, step **(3)** outlines objectives for deployment to a new user - that the system must adapt to these users, their configurations, and potentially brand new tasks. Reduced complexity allows better usability on commodity systems and interpretability allows users and experts to validate the models learned processes.

CHAPTER 2

BACKGROUND

This chapter discusses the prior work and concepts necessary to motivate and support this dissertation. The contributions presented in later chapters are all ML-based methods - methods in which the approach tunes or approximates a solution without human guidance, typically using data or simulated environments. ML is one of the primary approaches to implementing modern notions of AI, which are systems that behave with at least the skill-level of a human, but require no human assistance. Therefore, in this work ML and AI will be nearly synonymous: ML will be used when referring to practical approaches that learn from data, while AI will be used more generally to refer to the potential universe of automated solutions that don't require humans.

We begin the chapter with an outline of the recent efforts to define and establish *trustworthy* AI in Section 2.1. Aspects of trustworthy AI that our methods contribute within ML approaches - interpretability, privacy, and usability - are highlighted in more detail. We then introduce HAR in Section 2.2, beginning with a brief overview of human-machine interaction and continuing with a focus on signals collected from bioelectric sensors and their underlying challenges. In Section 2.3, we review fundamental concepts of deep learning, the primary ML method used in our contributions. Our final background section, Section 2.4, discusses ML methods related to our work on improved trust and adaptability relevant to this dissertation. Chapters 3, 4, and 5 include detailed discussion of our contributions' relevant methodologies.

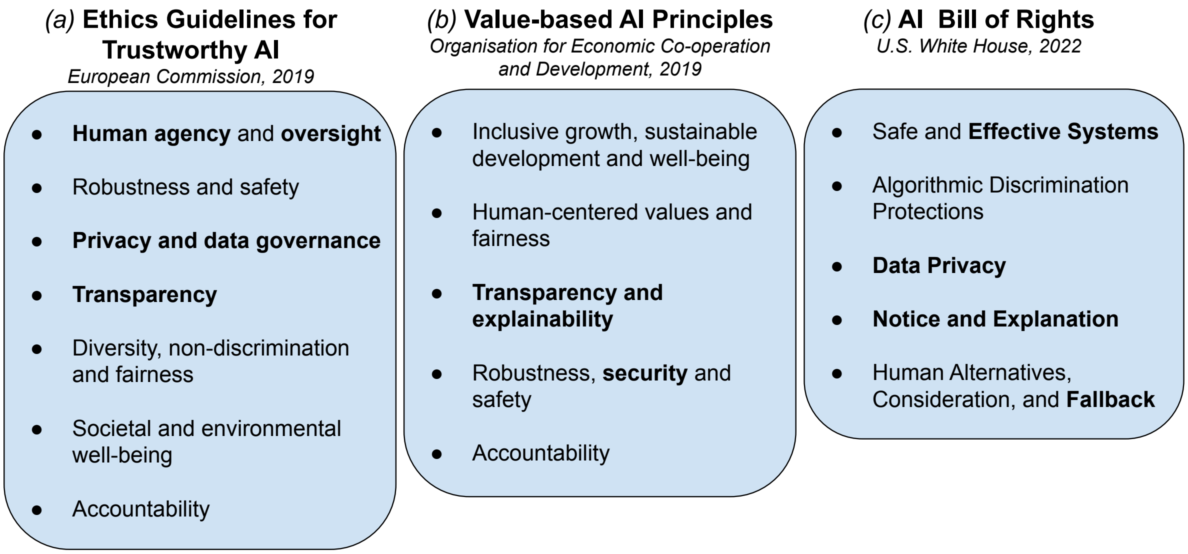
2.1 Trustworthy Artificial Intelligence

It is common to make decisions based on the assessment of someone or something else - whether it be a street light or an investment broker, the delegation of critical tasks is a cornerstone of everyday life. We *trust* these external systems to help manage tasks, often implicitly through our choice to use them. As many likely experience, higher risk scenarios may require increased trust in whatever is providing the assistance [43]. In addition to more clear requirements, such as having high accuracy, trust is also made up of less clear drivers, like the ethics or social norms it should abide [44]. However nebulous, these requirements of trust can be met, of course, by autonomous computer systems - a human element is not a prerequisite for other humans to form trust. For example, a person who is blind trusts their guide dog and a person with movement disabilities may rely on a cane or wheel-chair.

The importance of trust in computer systems was recognized early in the modern era by the national academies [45]. Today, advanced autonomous solutions are now commonplace, with growing concerns about their ability to manage risk or their potential to cause direct harms. For example, some researchers argue that delegating work to AI is a systemic risk that threatens to destabilize economies [46, 47]. Work has also highlighted the risks with generative AI, including its potential to insert malicious code when utilized for code creation [48]. Researchers also illustrated the negative opinions on the trustworthiness of delegating research tasks to an AI [49]. Others have offered potential regulatory frameworks for their use in high-risk applications by decomposing the challenge across the roles in the "value chain" [50].

In order to characterize trust, researchers consider the factors behind the decision to take on risk at the discretion of another, possibly autonomous, system [51]. For instance, systems may be reliable - i.e., performing correctly, reporting confidences, etc. - but this does not grant trust from the user. Other factors, such accountability and performance likely also influence our formation of trust. Improving these aspects makes a system more *trustworthy* - defined as the ability to have a "*firm belief in the reliability, truth, or ability of someone or something*" [15]. Under the frameworks in [43], an appropriate level of trust is established through assurances indicated to users. They argue that much of trustworthy AI research is actually oriented towards improving these indicators of trust. Therefore, trust is context and assurance specific, and will vary between users, and therefore should be measured through Trust-Related Behavior (TRB) - behaviors indicating a willingness to delegate risk to the AI. It has also been recognized in [52] that trust of autonomous systems is entangled in how it relates to various institutions, such as regulators, vendors, and users - and that clear steps are needed to begin establishing the complex relationships that will support trustworthy AI.

The growing use of AI systems has motivated several governing organizations to propose principles of trustworthy and ethical operation of such systems. These efforts are forward looking, attempting to help guide future endeavors towards results that share today's secular principles. While some existing laws do address information privacy [53, 54], proposals related to AI trust recognize that AI and automated systems bring new challenges. Figure 2 highlights three recent such proposals, which we briefly discuss in chronological order.



(a) **Ethics Guidelines for Trustworthy AI**
European Commission, 2019

- **Human agency** and **oversight**
- Robustness and safety
- **Privacy and data governance**
- **Transparency**
- Diversity, non-discrimination and fairness
- Societal and environmental well-being
- Accountability

(b) **Value-based AI Principles**
Organisation for Economic Co-operation and Development, 2019

- Inclusive growth, sustainable development and well-being
- Human-centered values and fairness
- **Transparency and explainability**
- Robustness, **security** and safety
- Accountability

(c) **AI Bill of Rights**
U.S. White House, 2022

- Safe and **Effective Systems**
- Algorithmic Discrimination Protections
- **Data Privacy**
- **Notice and Explanation**
- Human Alternatives, Consideration, and **Fallback**

Fig. 2. Outline of three recent frameworks for trustworthy and ethical Artificial Intelligence (AI). In bold are areas in which our work contributes, these general areas include: *interpretability* (Section 2.1.1), *usability* (Section 2.1.2), and *privacy* (Section 2.1.3)

(a) Seven essentials for achieving trustworthy AI [55]: The European Commission established an independent High-Level Expert Group (HLEG), which published their report in April of 2019, after receiving input from over 50 experts in AI, civil society, and industry. The group recognizes the wide range of issues globally that AI is poised to help solve, but found a need for guiding principles to help ensure well-being and welfare of the people. The report first recognizes three foundations for trustworthy AI - that AI be *lawful*, *ethical*, and *robust*. The HLEG expanded on these foundations with seven key requirements for trustworthy AI, as shown in Figure 2(1) and discussed by a member of the group in [56].

Table 1. Summary of trustworthy principles and their associated perspectives [15]. In bold are areas in which our work contributes, these general areas include: *interpretability* (Section 2.1.1), *usability/availability/autonomy* (Section 2.1.2), and *privacy* (Section 2.1.3)

Perspective	**Principles**
Technical	**Accuracy**, Robustness, **Explainability**
User	**Availability**, **Usability**, Safety, **Privacy**, **Autonomy**
Social	Law-abiding, Ethical, Fair, Accountable, Environmental-friendly

(b) Value-Based AI Principles [57]: An inter-governmental body, the Organisation for Economic Co-operation and Development (OECD) was founded in 1961 to aid in evidence-based global policies and standards. The group proposed both the five *Value-based AI Principles* highlighted here, as well as *Recommendations for Policy Makers* that adhere to the aforementioned principles when enacting governing rules. The OECD's principles are specific to AI, and are seen as complimentary to previous reports addressing topics such as responsible business conduct and data privacy.

(c) AI Bill of Rights [58]: The *United States White House Office of Science and Technology Policy* published the *AI Bill of Rights* to guide the "*design, use, and deployment of automated systems to protect the American public in the age of artificial intelligence*". The office developed this framework through input from industry stakeholders, technology developers, policy makers, and direct input from the public and impacted communities. Though not a legal document as the name suggests, the report outlines five principles along with a guide titled *From Principles to Practice* for those seeking to align with these ideas in the field.

As regulatory frameworks emerge to guide the lawful development and use of AI, researchers and developers must work to meet the demands of trustworthy services when seeking broad adoption. Recent work distilled aspects of trustworthy AI into three different perspectives: ***technical***, ***user***, and ***Social*** [15]. The principles that fall within each of these perspectives are show in Table 1 and are clearly linked to the governmental principles illustrated in Figure 2. The following subsections provide further examination of the trustworthy components we aim to address in this dissertation.

2.1.1 Model Interpretability and Transparency

All surveyed frameworks for ethical and trustworthy AI include core principles in support of autonomous systems that are *interpretable* and *transparent*: systems that provide some rationale behind outputs and their operations for producing outputs [18, 17]. Sometimes known as Explainable Artificial Intelligence (XAI), prior work recognizes that interpretability is primarily motivated as a means to improve trust in autonomous systems [59, 60]. Transparency as applied to autonomous system is more wide-ranging, often with interpretability as a sub-factor, and relates to deobfuscating details of the system in order to build trust in a certain cohort [61]. In this section, we first define interpretability and its aspects before considering how it relates to model transparency. We conclude this section with areas this dissertation contributes to trust through interpretability.

As noted in [62], terms like *interpretability*, *explainability*, *intelligibilty*, and *understandability* have been used almost interchangeably in ML literature. However, *interpretability* and *explainability* are the two key terms that appear most common in recent work. One distinction, borrowed from Cynthia Rudin's position paper [63], is that an interpretable model is inherently interpretable by design, while an explainable one requires additional analysis to be performed after the model is constructed in order to get an explanation of processes. More qualitative discussions in [64] argue that terms such as interpretability and explainability are poorly defined, but the field should focus on *the concept of helping a human understand something* as being interpretable. In their framing, *explainations* are simply the "*...currency in which we exchange beliefs*", a definition borrowed from psychology. Understanding allows people to form beliefs and consider other abstract concepts, such as biases or concerns about privacy. Having motivation to gain an understanding implies a lack of understanding, or an *incompleteness* of available descriptions. For our work, we do not draw a specific distinction between terms such as interpretability and explainability. Instead, we consider these terms synonymous for [64]'s description of simply helping people to understand how an ML model produces its outputs. In the remainder of this section, we'll review research to better understand the dimensions of interpretability as it pertains to our work.

Work in [60] recognize a trade-off between level of interpretability - or human understanding conveyable from "interpretable" descriptors - and the completeness of the explanations. As the descriptors become more complete, they more faithfully represent the underlying processes. However, these more complete descriptors meant for interpretability become more complex to comprehend, decreasing the interpretability they are intended to provide.

For different tasks and systems, a solution may be interpretable either globally or locally. In other words, approaches vary in *what* the interpretable method is able to describe in terms of sample locality or specificity. A ***globally interpretable*** method is able to describe how it operates in general for the inputs it receives. It is able to describe its broad rules and considerations when determining outputs or decisions. In contrast, a ***locally interpretable*** methods is able to describe how it arrived at a specific output for a specific input. It does not have to describe all steps or considerations, but descriptions should be specific to the input and reduce incompleteness. [64]

How models are explained can be categorized into ***explainability by design*** and ***post-hoc explainability*** [22]. Models that are explainable or interpretable by design, sometimes referred to as ***integrated interpretability***, require only an analysis of the model's structure or learned parameters. However, this can limit the complexity of the model, and likely impact the model's performance [59]. Post-hoc methods instead typically analyze inputs, intermediate results, as well as outputs to aid in the interpretability of "black-box", opaque models [22]. When developing solutions, the decision to rely on integrated or post-hoc explanations is not always clear. For example, practitioners seeking trustworthy solutions may find that simplified architectures require complex feature engineering for predication performance, making them still difficult to interpret. The performance shortcoming of integrated interpretability highlights the need for post-hoc predictions of complex models applied to unaltered input features [65].

Interpretability is valuable for trustworthiness within many contexts. For instance, when the system is highly accurate, it may still help establish trust to review it's criteria for insights into the underlying processes [65]. Understanding that the model is utilizing correlations and mechanisms previously validated as predictors can improve trust. However, even incorrect predictions can maintain trust by providing a reasonable explanation for their output. Interpretability can also enable negatively impacted individuals - people who feel that they are being unfairly disadvantage by the system - to question, challenge, and possibly remedy outcomes [17, 18]. Interpretability is not always required - perhaps ideally only in low-risk or well-understood problems should the lack of interpretability be tolerated [64]. In some cases, it may seem reasonable, given that a model is highly effective and has earned trust through demonstration, that interpretability is no longer important [65].

Transparency is often related to interpretability, but with increased scope and recognition of social mechanisms. Transparent systems are expected to disclose any aspect that supports a model - including data collection, data storage, and attestation of regulatory compliance [22]. However, transparency is not *always* desirable, and may degrade trust in a system [61]. These properties arise as the people involved, their motivations, and their contexts change. Ultimately, who benefits and what information the transparency provides primarily determines the impact of transparency. Underlying many assumptions of transparency being desirable is the assumption that the information is accurate. However, earlier work has even offered that scenarios in which deception may improve trust [66]. For example, mutually beneficial lies might increase trust from others whom considered the lie benevolent due to the outcome it generated.

Our work focuses on model interpretability, regardless of whether the interpretation is made transparent to an end-user, to experts only, or just used by a developer to verify their work - we assume that interpretability improves trust in AI solutions. In Chapter 3, we contribute two separate methods for interpretable feature extraction from neural signals. Both methods are globally interpretable, integrated solutions. We apply a similar approach in Chapter 4, but within a transfer learning paradigm applied to Myoelectrocortography (EMG) data, illustrating to experts and users the change in frequency spectrum when adapting to a new user.

2.1.2 Model Usability: Availability, Autonomy, and Reproducibility

Trustworthy modeling systems are expected to achieve *usability*. Concepts of model usability are broad, but focus on the capabilities afforded to the user, and any conditions restricting use. Properties of usability include *availability* and *autonomy* - that model systems should be easy to use, and prepared to produce output whenever they are needed [44] [15]. The ability to interrogate a model or autonomous solution - assessing its operations closely - is important for building trust [61] with other developers and experts. Finally, it has long been understood that usability and security are often at odds - increased usability implies a reduction in the restrictions that may aid security [45].

Importantly, there is no clear formal definition of usability, and instead may be highly context dependent, relying on expected social norms [44]. For our work, we define usability as the ability to execute and produce predictions locally, on systems the user controls. Approaches that instead require data be sent to a third-party in order to receive output are limited in their usability. In these cases, users are required to stay connected to the internet, or other local network, preventing usage in environments lacking access or connectivity. Trust is degraded because availability and autonomy are reduced. Should the AI provider cease doing business or otherwise cease maintenance, the solution is also no longer usable. Even network or other infrastructure outages will dictate the use of the solution and degrade trust. These restrictions are not required to be network related - for instance, an intelligent system may be distributed on provider-owned hardware, with restricted access. It may be a more lucrative deployment option for the provider, but trust is degraded because the solution cannot be easily maintained or transferred without continued engagement with the provider. Even a trustworthy provider cannot guarantee its continued existence or support. These issues are obvious for predictions, but they also apply to any training process required to adapt to a user's own data. The solutions are made more trustworthy by not requiring transferring data to external systems. Usability can be encouraged and supported through reasonable conditions, such as government incentives and security protocols that maintain privacy, and training procedures that are reasonably short and infrequent.

Given these desires for improved usability, any AI that takes input from human bioelectric signals should aim to be usable on local systems. Regardless of autonomy needs, many applications of activity recognition are not well-served by high-latency and low-bandwidth remote systems. In many practical applications, such as a muscle controlled smart prostethesis, higher-latency response to the user's intent would limit their reaction time and overall mobility. However, even if these constraints do not entirely compromise functionality, they still undermine trust through degradation of usability.

When considering the full life-cycle of AI systems, the end-users' usability is also an issue of *reproducibility*. While reproducibility is often thought of in terms of reproducing research experiments, model reproducibility clearly requires the ability to produce new output from novel inputs to a trained model [22, 23]. The domain of ML has recognized this aspect of reproducibility, with methods to address common issues, such as code, data, and model availability [23, 67, 68].

Our contributions to usability are solutions that aim to reduce the complexity of model execution and adaption. Reducing complexity reduces computational costs, and therefore the burden of model building and usage is reduced for individuals. In Chapter 3 and 4 we use engineering-informed approaches to reduce model complexity, towards methods that are readily executable on low-power commodity devices. In Chapter 4 and 5 we develop methods that do not need to be re-trained "from scratch" for a new participant, simplifying the model's adaption at deployment for improved usability.

2.1.3 Data Privacy

The requirement of *data privacy* is a key pillar in most frameworks for trustworthy AI [55, 58, 15]. To understand the practical concept of privacy, we first briefly consider its history. We then discuss the model based attacks and related concepts in support of later chapters.

The notion of the *legal* right to privacy is considered a fairly recent human invention. It was first identified as a *common law*, inherit through custom and precedent, that an individual had the right to choose whether to share their "private life, habits, and relations" [69, 70, 71]. Over time, legal frameworks developed further [72, 53, 54], with early focus on health science's balance between privacy risks and research benefits of individual health information. Under this context, an individual's private health or behavior data (sometimes called "microdata") can be combined and then provided to researchers for studies that benefit science and society. But scientists recognized that the sharing of private data brings a risk of violating the right to privacy for those contributing data. For these reasons it is common practice to de-identify data when sharing samples for research purposes [73]. This practice aims to reduce the risk of ***re-identification***: the ability to use seemingly anonymous data to determine the particular individual associated with that data. Re-identification is important because it can be a natural first step towards further privacy violation. The risk of re-identification can be measured as the likelihood that a sample can be linked back to the individual given the data [74, 73]. Considering the level concern for information privacy, significant work exists across many fields to help ensure it.

The survey of work in [75] highlights that the success of the *big data* era comes at the increased risk of privacy for individuals. In order to realize all the benefits of increased information, individual expectations of ***content privacy*** and ***interaction privacy*** must be met. They propose that methods should meet each user's own privacy expectations when using information relating to the individual (*content privacy*) or when communicating information on behalf of the individual (*interaction privacy*). It is intuitive that content privacy can be trivially achieved in many research and application contexts - simply assume no utility in a shared database of information and only develop processes that require a user's own data on their own systems. Similarly, interaction privacy can be achieved by simply not interacting [19]. These standpoints are impractical, but they demonstrate through extremes the balance between utility and privacy that must be considered.

ML systems require data for both training and run-time production of outputs, meaning that ML developers must also face privacy challenges. However, compared to the more narrow concern of statistical tools, many new dimensions of privacy are recognized in the broader applications of AI. The National Institute of Standards and Technology (NIST) outlined a taxonomy of Adversarial Machine Learning (AML), which they describe as the study of algorithmic security challenges, attacker capabilities, and attack consequences [76]. Many aspects of their taxonomy are related to the privacy of sensitive information within original training data, but they also recognize other risks, such as poisoning attacks that shift decision boundaries or reduce accuracy but do not necessarily impact privacy. Outside of attacks requiring training data access, they also recognize challenges such as the development of evasion techniques that use small data perturbations to invoke an erroneous output. *Adversarial samples* such as these don't directly threaten the privacy of the training data, but they are a flaw in the reliability, and therefore security, of a solution.

Information *leakage* facilitates re-identification and other attacks against a model and the data used to train it. The information was intended to be hidden or secure from an attacker, but through some means has been *leaked* to the attacker. Modern notions of how information is leaked were described recently in [18]. ***Direct information leakage*** occurs when someone gains access to samples, such as an attacker gaining access to a database of training samples containing sensitive information. Information security controls and other information technology best practices are fundamental to preventing direct information leakage. ***Indirect information leakage*** occurs when, *without* access to all training samples, an attacker is able to use inference algorithms to recover information originally contained in the training data. The strength of privacy can be measured through simulation of various threat models, typically with the aim of demonstrating feasibility or a likelihood that an attack can succeed with the level of information leaked. For example, we can assess the risk that an attacker could infer that a sample was in the training dataset by simulating a membership inference attack [25].

Researchers in [77] have identified a trade-off between model fairness and model privacy, both considered pillars within trustworthy AI. The issue arises from the techniques typically used to adjust unfair models to become more fair. When ML systems under-perform on under-privileged cohorts, techniques enforce constraints on the learning process in order to better support the under-privileged cohort. The results help to equalize performance, but experiments demonstrated they also increased the risk of membership inference attacks. Larger biases, and therefore increased need for fairness, were also correlated with increased privacy risks.

Based on the previously highlighted trustworthy AI frameworks, our work assumes that trust is improved if the data required are kept sufficiently private for the application and user tolerances. Therefore, how training and run-time input data are processed helps determine the level of trustworthiness of a machine learning system. For example, requiring the user's data be sent to external systems undermines trust in the system because data is no longer private. The external third party may not store the data securely, and it may be sold or otherwise reused for purposes the user did not originally intend. The user may not even know the form of the data captured. Intuitively, these privacy needs overlap with trustworthy requirements surrounding *usability*, described in the previous section, and therefore both motivate less complex systems usable on commodity resources.

Contributions in Chapter 3 and 4 help ensure data can be processed locally on systems the user owns and trusts. In Chapter 5.3 we propose two AML experiments with our brain2vec transfer learning approach. One experiment is a straighforward assessment of re-identification risk, and the second a more nuanced experiment for the risk of membership inference attacks.

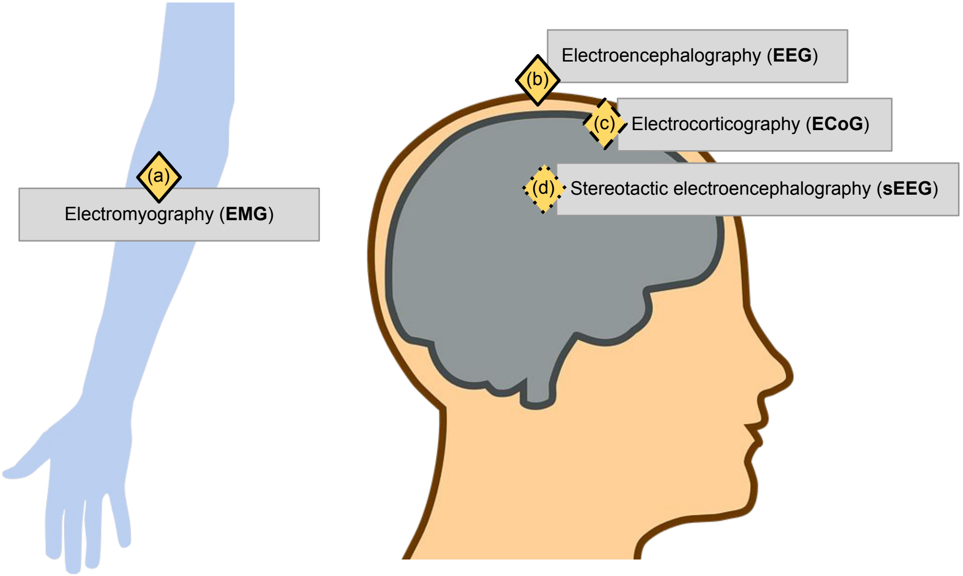

Electroencephalography (**EEG**)
(b)
(c)
Electrocorticography (**ECoG**)
(d)
Stereotactic electroencephalography (**sEEG**)
(a)
Electromyography (**EMG**)

Fig. 3. Bioelectric sensor modalities used in this work: **(a)** surface Myoelectrocortography (EMG) captures electrical activity from electrodes placed on the skin surface, produced from the contraction of skeletal muscle sarcomeres. **(b)** scalp Electroencepahlography (EEG) records bioelectric activity of neurons, diffused through skull and scalp tissue, from electrodes placed on the scalp. **(c)** electrocorticography (ECoG) measures neural activity from electrodes placed directly on the brain surface (cortex), and **(d)** Stereotactic Electroencephalography (sEEG) measures neural activity from deep brain structures from depth electrodes implanted via stereotactic guidance.

2.2 Humans and Computers

In the following subsections, we explore how humans use machines and how their use has evolved over time. We first highlight the history of research on HCI, tracing this foundational research to the work in the HAR domain being performed today. Our final subsection discusses HAR specifically and the relevant challenges for this dissertations focus.

2.2.1 Human Computer Interaction

HCI is a domain of research regarding the design, evaluation, and implementation of interactive computer systems for humans. Its scope includes both HCI as a tool to better communicate an individual's desires to a computer system, as well as a tool to study human phenomenon and physiology [78, 79, 80]. The concept of HCI, with overlapping research in Human Machine Interaction (HMI), is related to Human Robot Interaction (HRI) in which researchers focus on robotics. Early worked used the term HMI to discuss the interactions that occur as a person attempts to satisfy their intention [81]. Their *stages* of interaction were as follows:

1. User forms the intention

2. User selects an action
3. User executes an action
4. User evaluates the outcome

From these steps it is clear that HCI is fundamentally user and task centric. If a task - the goal of each step above in a particular application - can be automated, without a human component, then HCI concerns and designs are irrelevant.

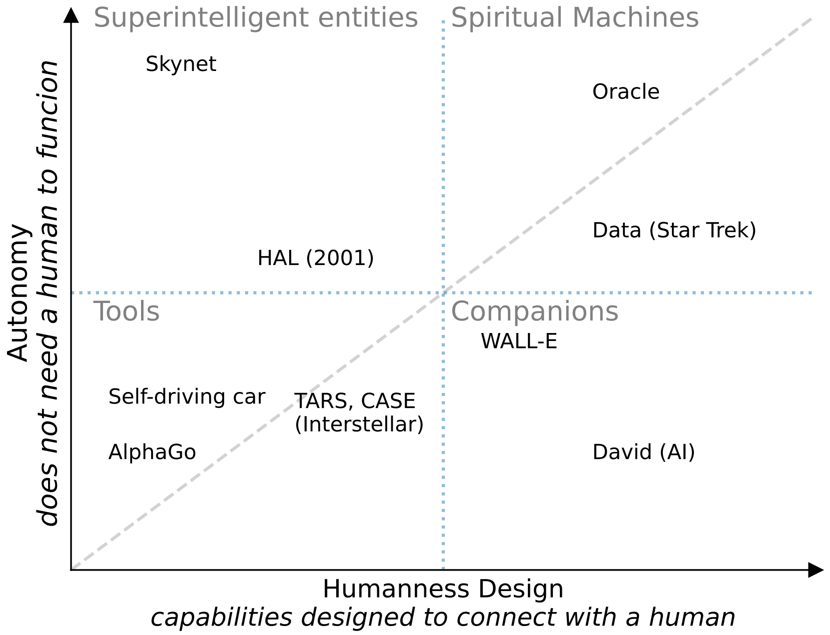


Fig. 4. Based on [82]: the dimensions of autonomy and humanness. High autonomy improves the expected utility across domains, while high humanness improves potential for human-machine trust.

Early computer interfaces progressed primarily through university work that necessitated an interface to these new systems, but development quickly expanded to industry as the potential became more clear [80]. Remarkable progress erupted with digital computing and the development of displays, keyboards, and mice. Today, the domain continues to develop with more recent solutions such as mobile text-to-speech and AI assistants. Modern low-power systems have enabled more pervasive implementations that can passively monitor human behavior in order to monitor and gauge a person's status [78, 83]. Research on HCI recognizes the notion of a *user interface* and the obvious need for *usability* - so much so that some predicted the end of the user interface by the year 2020 [80]. Technologist even expected machines to anticipate human needs, enabling more seamless use.

Part of the increased expectations for HCI, as discussed in relation to AI in Section 2.1, is the demand for mobile systems that are useable anywhere, with few restrictions. More recent research has recognized the need for *trust* and *trust repair* in human-machine interaction, proposing that increased machine autonomy must be met with a human-centered approach to trust. The simple first step proposed by the authors is to begin treating the interaction as being between two humans, rather than a human and their tools. Under this framework they consider the important of trust-repair in human-machine relationships [82], which may need to use emotion recognition solutions to gauge trust [84]. The authors of [82] call on researchers to rethink the connection between humans and their technology. They propose framing the relationship not as human-to-machine, but as another human-to-human interaction. From this assumption, they consider the *humanness* of a technology's design as it relates to its level of *autonomy*. We adapt a figure from their work relating the dimensions of humanness and autonomy in Figure 4. They argue that increase humanness is needed if a strong bond and communication with the human user are required.

Included in HCI is research on monitoring for scientific research and human health purposes, which often overlap in needs with interfacing for everyday tasks. Brain-Computer Interface (BCI) and other bioelectric signals, such as EMG are often recognized and utilized as potential low-cost and portable solutions to a variety of HCI problems and tasks [80]. More recent material's research has highlighted the value of wearable sensors for HCI, calling wearable sensors an "*inevitable future trend*" [85]. Similar wearable sensors are a common method for enabling human exoskeletons - robots that closely interact with humans to augment their capabilities. Sensors and their processing systems monitor the environment, including the user, in order to help control the exoskeleton [86].

HCI has a broad focus on enabling humans to interact more successfully with their technology solutions. This includes comprehending the user's intentions and actions, as well as interacting with the user and the environment. The domain of HCI leverages new technologies to improve the human experience of the interaction, including advances in AI and bioelectric sensing. This work's contributions focus on comprehending the user's intent or action, and would fit well within any future HCI solution. We argue that, given HCI's history and expected trends described in the above discussion, there is significant overlap with the field of HAR, which we further discuss in the next subsection.

2.2.2 Human Activity Recognition from Bioelectric Signals

HAR enables rich user experiences for many applications, with the potential to improve the livelihood for persons with disabilities [87, 88]. HAR solutions may use a variety of different data modalities to recognize human action, including video, audio, accelerometer, and biophysical data [7, 8]. The solutions that HAR provides are clearly linked to the domain of HCI: HCI implies a human interacting with a machine through some action, and HAR focuses on approaches that allow machines to comprehend the action a person is taking or their current experience. We therefore view HAR as the solutions that *read* the human experience, a sub-component of HCI, which is a broader domain that includes response and actuation of the environment.

Each of the steps listed in Section 2.2.1 may be considered an "activity" relevant to the HAR domain. In other words, each step is a plausible target for automation or assistance using HAR approaches. A systems may predict or detect the users intention's and their desired action (i.e., step 1 & 2 - selection & attention). The system may detect the user's execution of the action or even help them perform their desired action, perhaps automatically (i.e., step 3 - execution). Finally, a system may use the individuals physiological response to gauge their satisfaction with the outcome (i.e., step 4 - evaluation). Depending on the user's needs, HAR methods can reduce overhead of these steps in order to improve the experience of HCI contexts.

This work's focus is on time-varying bioelectic data captured to measure human response. Unlike video monitoring for activity recognition, using sensor readings is often lower cost and pervasive, without violating the privacy of individuals nearby. However, many challenges in HAR remain.

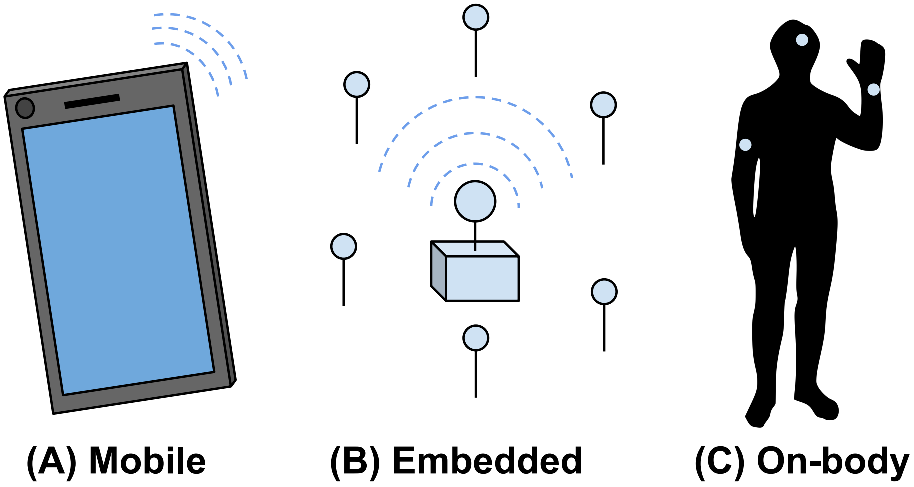


Fig. 5. Examples of resource constrained platforms often used to implement activity recognition from sensor signals [16]. On-body sensors are used to capture bioelectric signals, but they may be processed by nearby mobile or embedded devices.

Resource Constrained Platforms: Data are collected through research trials targeting the data (e.g. expression of activities of daily living [41]), clinical proceedings tangentially related (e.g. monitoring response to medical procedures [89]), crowd-sourced from interested users (e.g. Common Voice by Mozilla [90]), or simply harvested from a existing user-base (e.g. mobile applications). Sensor-based HAR approaches often rely on mobile or Internet-of-Things (IoT) [83] devices equipped with sensors continuously monitoring the subject. In order to expand practical use-cases, recognition using this data is performed on-device, which minimizes response time and eliminates reliance on external systems [91]. However, mobile host systems are typically resource-constrained, often requiring low power and reduced weight in comparison to traditional desktop or server host systems. Thus, activity recognition models are motivated to reduce model complexity while still maintaining acceptable recognition performance.

Resource constrained applications demand low-power and low-latency operation to be viable. In contrast, problems with fewer constraints - perhaps only capital budget constraints - are permitted to utilize as many large server class machines as can be afforded. In these cases, issues relating to amp-hour usage and weight of the system are considered in aggregate against organizational capability. In this context, resource constrained system exist to support a specific application function. For example, a step tracking device must be light enough to wear for the duration of tracking, with enough power to measure a day's worth of data. In contrast, a computing cluster built for optimizing and executing large machine learning models as a service will only need to comply with common civil and electrical engineering guidelines. The intended application determines the constraints, and the constraints of sensing human bioelectric signals are typically strongest along dimensions of power usage and weight. Chapter 4 presents our work that reduces the complexity of on-body classification methods without reducing classification performance.

Distribution Discrepancies: A core goal of machine learning, and even statistical analysis, is to generalize to more than the distribution of their training data - a recognition that variability is inherent [92], and that generalization is a key issue [93]. Methods are expected to extrapolate to a reasonable extent, but large drifts from training to testing data can reduce the scope and overall utility of machine learning methods. The HAR domain experiences these discrepancies due to the varying physiology and behavior of individuals. We illustrate this challenge in Figures 6 and 7 with data used in our contributions. In these examples, we can see clear similarity across participant's and their sensors, yet large differences likely to complicate analysis also exist. Bioelectric signals are further complicated by their own challenges, such as sensor reliability, electrical interference, and physical placement [5]. In Chapter 4 and 5 we implement solutions that transfer knowledge in an attempt to overcome distribution discrepancies.

Readability: Bioelectric sensor systems capture information that is less familiar to humans. For example, most users can review video or audio recordings used to classify an activity, and judge the system's capability in part based on their own ability to interpret the information. Researchers have argued that these other human-friendly modalities have inherit interpretability that is not present for modalities like sensor recordings [5]. Therefore, effort must be made to build interpretable solutions that experts and users may consider when using the system. We highlighted the value of interpretability earlier in Section 2.1.1. Our contributions in 3 and 4 both implement interpretable models that can provide insight into how the data is used for predictions.

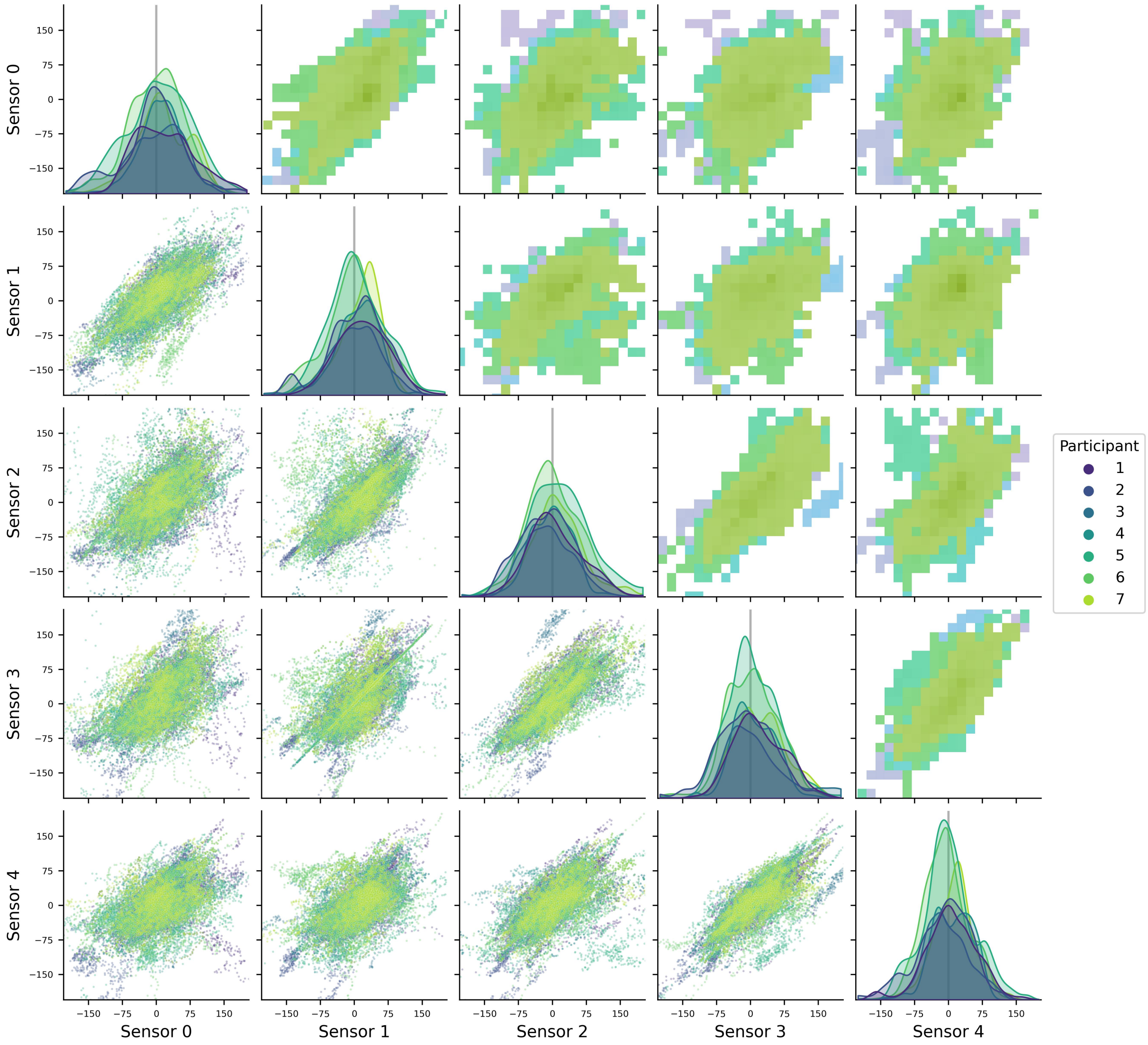


Fig. 6. Examples of distribution discrepancy demonstrated by a selection of sensors across different participants, taken from data described in Section 5.2.1. Expected values and their distribution vary widely between participants and even within a participants separate sensors. While all Machine Learning (ML) is intended to generalize across some amount of changes to the underlying distributions, the domain of human activity recognition is challenged by a persistent and seemingly inescapable drift across users and configurations. While real-time drift can also occur, for instance as a user becomes tired their muscle activity distribution may change, this work focuses on domain-level human activity discrepancy.

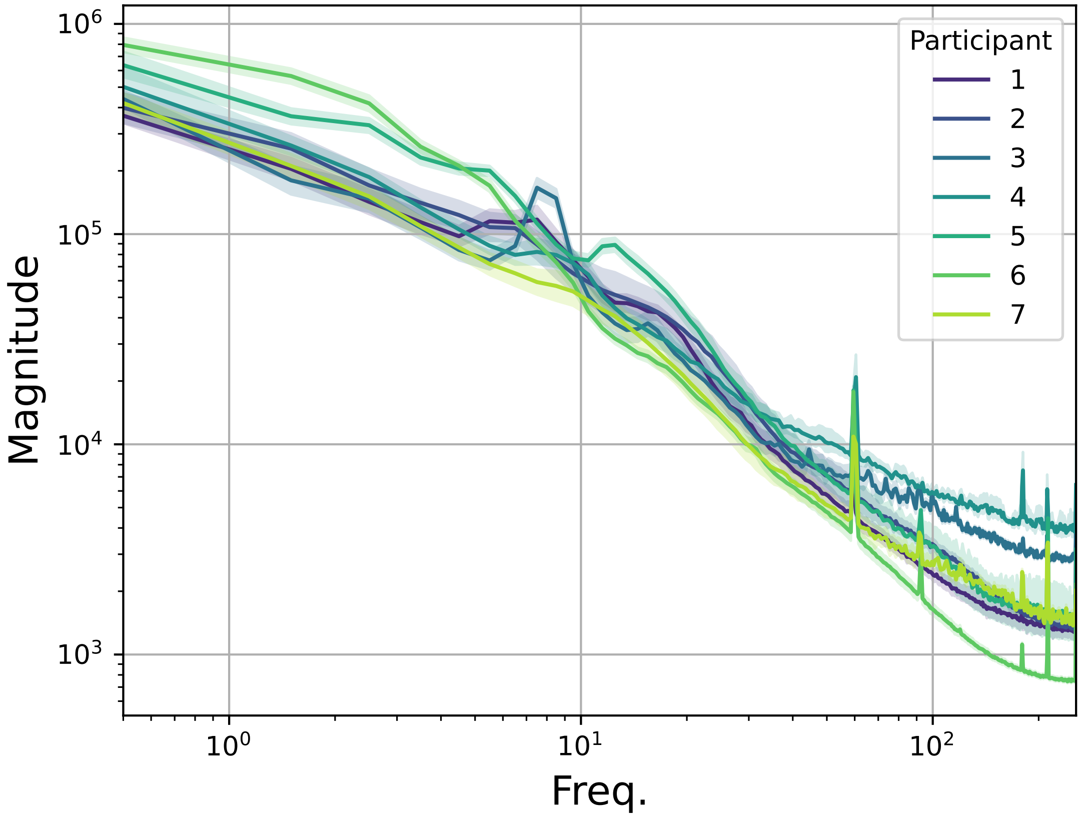


Fig. 7. Examples of distribution discrepancy demonstrated by the average sensor frequency response across different participants, taken from data described in Section 5.2.1. Higher values along the Y-axis correspond to a increased representation of the corresponding frequency on the X-axis. While each participant's data is similarly shaped, their still appears to be distribution discrepancies from shifts in amplitudes, new spectral modes, and varying noise levels across the participants.

2.3 Deep Learning

This dissertation's focus is to help establish ML techniques for human activity recognition from bioelectric signals that are both adaptable and trustworthy. ML describes techniques that automate the discovery of relationships within data, and to utilize those relationships in practical applications. More broadly, ML methods map from one distribution to another distribution using some criterion. [1] It is the developer or user that must define learning algorithms criterion to target a useful and worthwhile problem.

This work's focus is on more recent methods known generally as *deep learning* [28] [96]. Deep learning techniques attempt to *optimize* a well-formed problem, albeit with compromises and simplifications, in hopes of the model *learning* highly informative representations. Successful representations allow the model to be used for many separate tasks or simply out-perform other methodologies.

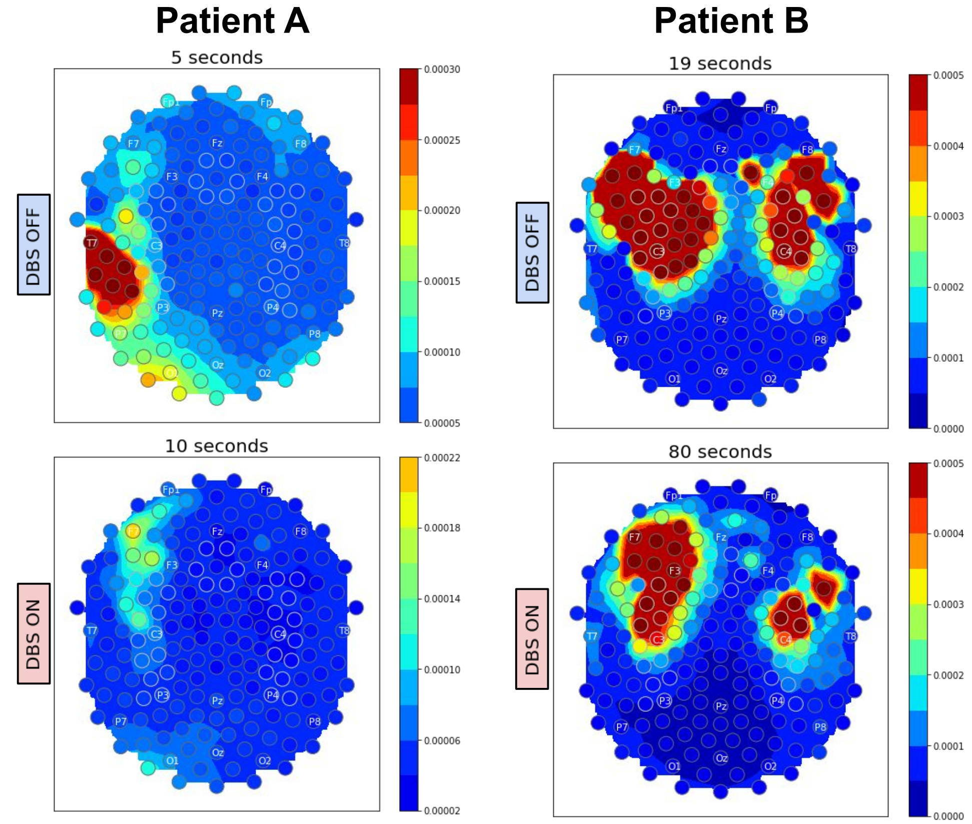


Fig. 8. Example of distribution discrepancy of a domain-specific measure: An illustration of phase amplitude coupling for two patients' responses to deep brain stimulation as measured by Electroencepahlography (EEG). This specialized heuristic (Phase Amplitude Coupling [94][95]) for measuring brain activity provides possible clinical insight, but still produces large discrepancies between users. Models and inferential statistics are challenged by these large shifts in a high-dimensional dataset. This analysis was performed as part of exploratory work following efforts described in Section 3.2.

2.3.1 Concepts and Components

Deep learning stemmed from artificial neural networks, which are ML methods that attempt to mimic biological neural systems: each artificial neuron is connected to many other neurons via synapses, with each neuron *firing* (producing output) based on the inputs received from other neurons.

In early work on neural networks, authors described the notion of ill-formed problems versus well-formed problems. An ill-formed problem, such as facial recognition from an image, does not have a known process and is possibly subjective. A well-formed problem in contrast is defined ahead of time or is a well-known operation, such as matrix multiplication. Therefore, a reasonable approach to ill-formed problems is to first decompose problems into these well-formed concepts and then proposing solutions to the well-formed aspects. The result, if successful, will lead to an approximate but useful solution to the ill-formed problem.

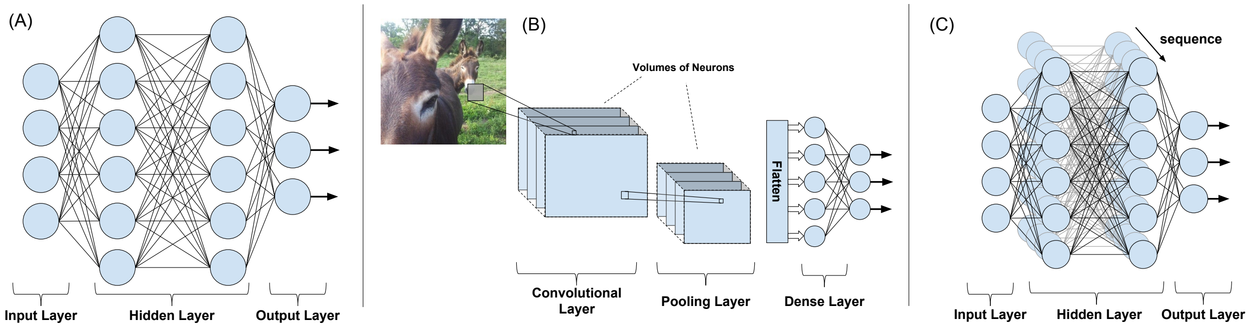


Fig. 9. Popular artificial neural network architectures: (A) Feedforward dense networks, (B) Convolutional Neural Networks, (C) Recurrent Neural Networks

Efforts in deep learning have further abstracted the artificial neural network model components into *layers* of non-linear transformations. Each layer accomplishes its transformation through *units*, a generalization of the artificial neuron. The term units is meant to expand the scope of the term to include newer, sometimes parameterized, components like batch normalization [96]. Still, the term "neuron" is used interchangeably with "units" throughout this and other research.

Artificial neural network models can be separated into many categories, but we begin with the three most dominant organizations. These are illustrated in Figure 9 as *feedforward*, *convolutional*, and *recurrent* architectures. These basic building blocks can be further combined in ways to accomplish more specialized learning strategies. One such recent development is the *transformer* network, discussed in Section 2.3.2, which combines several feedforward networks and learned transformations to better model sequence data.

A ***feedforward*** neural network is a type of artificial neural network that cascades values through layers of units. If values are transferred backwards through layers or across samples, the network is considered a ***recurrent*** neural network [96] [97]. Illustrations of feedforward and recurrent networks are provided in Figure 9a and 9c, respectively.

A ***convolutional*** neural network is a distinct organization of units, such that the layers implement n-dimensional filters. These convolutional layers are in contrast to *dense* layers that fully connect all units between layers. Given the properties of digital filtering, convolutional neural networks work well with highly correlated data and require fewer parameters than a fully-connected counter-part. When trained on image classification tasks, the resulting filters illustrate a hierarchy of 2-dimensional convolutional filters. Hidden layers deeper in the model, closer to the output, build more abstract filters capable of matching complex relationships. [98] [96]

Neural networks can be defined as a method of approximating some function - a mapping from one data domain to another - through a *training* procedure. In generalized terms, a network with parameters θ, input features x, and target variable y is described by the mapping

$$y = \mathbf{f}(x; \theta)$$

A traditional artificial neuron performs a weighted sum of its inputs and passes this value through an *activation* function. Formulation of a neuron receiving i features is given by:

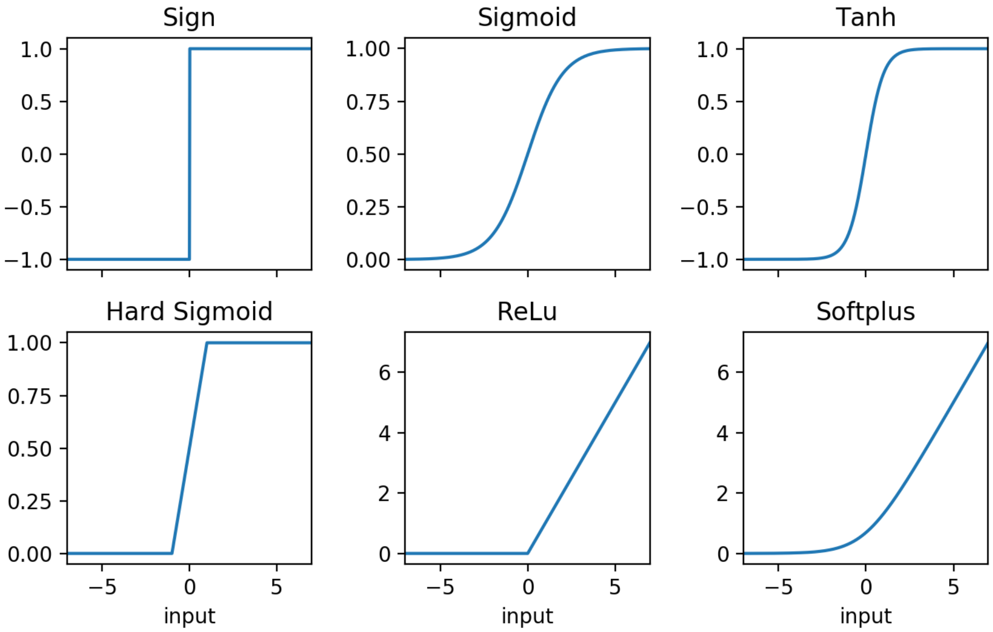


Fig. 10. Activation functions for neural networks. After weights and other operations are applied to a units inputs, the results are passed through the unit's activation function. Nonlinearity and differentiability are important characteristics of activation functions. However, more complex activation require more compute power, potentially limiting applications.

$$Z = \sum_i x_i \times w_i = \boldsymbol{W}^T \boldsymbol{X} \tag{2.1}$$

$$y = \phi(Z) \tag{2.2}$$

Where $\boldsymbol{X}$ is a vector of input features, $\boldsymbol{W}$ is a vector of weights or parameters, and ϕ is the neuron's activation function. This style of neuron is still at the center of most deep architectures' units, but these units often integrate other regularization or transformation operations.

Without a non-linear activation function ϕ, a neural network becomes a linear model, reducing the model's capacity to capture complex relationships [96]. A wide range of activation functions have been used historically, see Figure 10 for a sample of the more prominent non-linearities used for activation. In biological systems, neurons *spike* over time. Within this context, artificial activation functions in Figure 10 represent the *average* response of the neuron over time with respect to a sample. The choice of activation function can significantly impact both model accuracy and computational burden of training and prediction. Deep learning has pioneered the use of a Rectified Linear Units (ReLU), which helps to prevent saturation while also being fast to compute. [99] [96]

Training procedures attempt to optimize the network's parameters for the mapping $x \to y$ with a low error. Training is typically performed through a *cost* function, sometimes called an *objective* function, and *gradient descent*. The cost function represents the error in the network's output relative to the desired output. Minimas in the cost surface are pursued by calculating the error gradient and adjusting parameters to descend to lower regions of the cost surface. A key challenge with gradient descent applied to complex models like neural networks is determining the error gradient. *Error back-propagation* [100] is still widely used as a method to unroll the contribution of the error onto each neuron's parameters. *Mini-batches* of samples - subsets of the data that fit well within machine memory - are iterated over, and the gradient of each parameter, with respect to the output of the loss function, is calculated. The loss, or error, is minimized by using each parameter's gradient to update its value. An optimization algorithm adjust the parameter's value in the negative direction of the gradient, shifting the parameter in the direction that would result in smaller error. The process is repeated across every batch of samples in the training data. One iteration over the entire training dataset is termed an *epoch*, with most models requiring a hundred or more epochs to reach a performance plateau.

Deep learning methodologies typically have a large number of learned parameters. In order to optimize the parameters without overfitting, large datasets are required along with robust regularization. Furthermore, as training progresses, the magnitude of inputs and intermediate values play an important role in shaping the gradient. Large magnitude values push many activation functions into regions where the gradient becomes small. However, activation functions such as the ReLU, seen in Figure 10, help to address this by having a constant gradient for large positive values [99]. Another important consideration is the use of a weight initialization scheme that complements the chosen architecture and activation function [99]. Other methodologies to prevent *vanishing* and *exploding* gradients include feature normalization, batch normalization of intermediate layers during training, and sparsity regularization to bias weights towards zero.

This introduction outlined the basics of current deep learning techniques in artificial neural networks. The architectures and their layer types in Figure 9 are the basic building blocks of deep learning. Combining these architectures with various linear operations, branching data flows, and more thoughtful use of activation functions are the essential considerations of deep learning methods. From these methodological tools, new architectures that extend capacity, expressiveness, or efficiency across multiple domains emerge. Our work primarily applies feed-forward dense networks and different convolution methods to implement AI forHAR. We also rely on both sigmoid and ReLu activation functions in many of our contributed architectures.

2.3.2 Transformer Architecture

The *Transformer* represents recent progress in deep learning [101, 102, 103, 104, 105, 106], and we apply it in Chapter 5, but provide additional background in this section. A Transformer architecture, as illustrated in Figure 11, is made up of an *encoder* portion and a *decoder* portion. The transformer encoder first uses *self-attention* to learn how each element of the sequence relate to all the other elements, e.g., X_1 and X_2 in Figure 11. Self-attentions is a learned weighted representation of the inputs, enabling it to *attend* to any element in the input sequence. The inputs that the transformer can access are typically fixed, and often called the *context size* of the model. The output of the transformer's encoder is provided to each decoder layer, through another layer of attention, shown as the "Encoder-Decoder Attention" in Figure 11.

As introduced above, a key novelty of transformers is the method of **self-attention** [108] - a learned data transformation that relays information from any position in the input to any other position before further processing. This is performed by learning transformations over the input that result in a likelihood distribution that the network uses to weight input elements. For each d-dimensional embedding vector in the sequence X, a query, key, and value representations are extracted from an affine transformation with their respective learnable parameter matrices

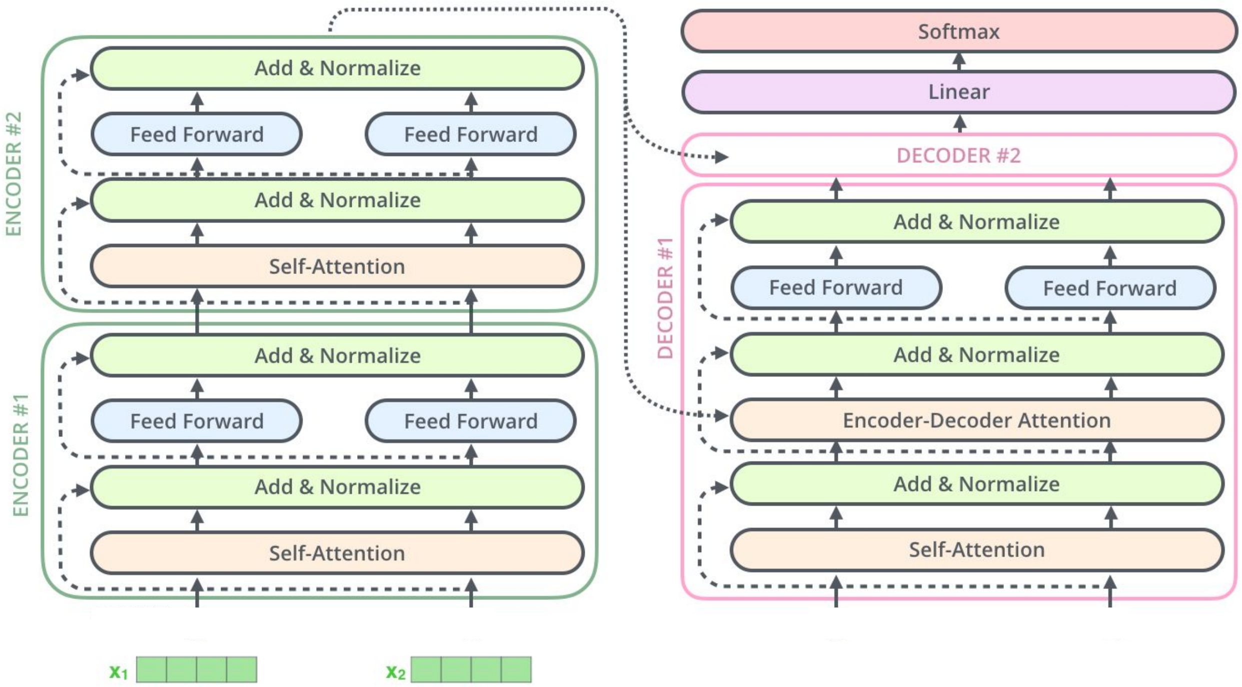


Fig. 11. The *Transformer*, composed of an *encoder* portion and a *decoder* portion, is used to model sequence data with *self-attention*. Figure adapted from [107].

W^Q, W^K, and W^V. The result of these transformations are Q, K, and V, respectively. The result of $Q \times K^T$ scores the hidden states and scaled to improve stability by $\sqrt{d_k}$, where d_k is the dimension of the key vector. A $softmax$ function is used to normalize the scores to sum to one - resulting in a weighting across all items in the sequence. This weighting is then multiplied with V to weight the contribution to the attention embedding Z for a particular item in the sequence. Each attention *head*'s Z is combined and transformed to a final output using a final learned transformation, with the weights W^O in Figure 12. As shown in Figure, 11, the resulting Z matrix is added to the original input X, then passed to the attention layer, and finally normalized.

While architected nearly identically, the encoder and decoder are differentiated by their inputs and the cross-attention layer. In their original work, [108] were targeting text translation when they developed the transformer architecture and its use of attention. In this paradigm, the encoder portion is provided the entire original text to be translated. The encoder extracts an informative representation that is passed to the decoder using the encoder-decoder attention layer. This way, the decoder can similarly attend to the portions of the encoded representations that are useful.

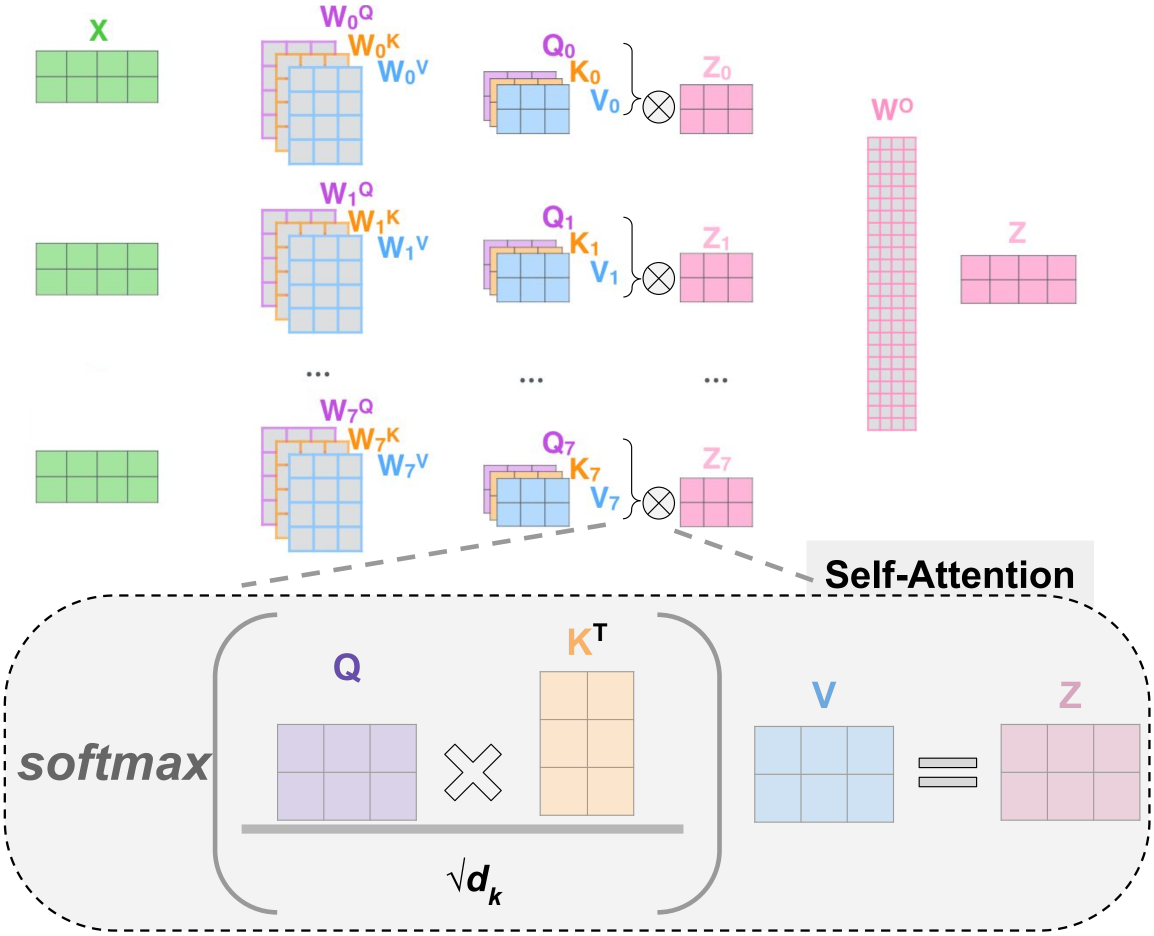


Fig. 12. Self-attention with 8 attention heads as used in [108]. Adapted from illustrations shown in [107].

2.4 Modeling Methodologies for Trust and Adaptation

This section discusses model methodologies that benefit trustworthy and adaptable AI for HAR, with a focus on deep learning approaches in ML. As discussed in Section 2.2, HAR models must contend with shifting distributions on lightweight mobile devices, making predictive performance and usability challenging. Of course, meeting any of the trustworthy principles described in Section 2.1 is desirable for a broad range of model applications, but the need for trustworthy models in the HAR domain might be considered heightened, given the domains personal nature of direct physiological sensing.

Deep learning methodologies are primarily defined by a choice of architecture, objective function, and a training procedure that utilizes the objective to improve the model using data. These various optimization inputs and decisions establish a *prior*, or applies inductive *bias*, to a model. Careful choice and definition of these priors can be used to predispose ML models to certain behaviors and desirable properties. An important desire is that the model generalize to variety of new data or even new problems. In other words, an important goal is to reduce the amount of inductive bias needed to create a new model for a new problem. This further relates to the bias and variance trade-off - models must be complex enough to capture the intricacies of the training data, but not so expressive as to overfit and poorly generalize.

In the following subsections, we discuss how engineering-informed model designs, like the ones this work contributes in Chapter 3 and 4, can make interpretable and parsimonious models for increased trust. Background is also provided on methods that allow models to establish, transfer, and extend prior knowledge in order to adapt to new tasks and domains. This dissertation contributes two methods utilizing these approaches, outlined in Chapter 4 and 5.

2.4.1 Engineering-Informed Architecture

For more narrowly defined applications, such as HAR from bioelectric data, the model architecture and training procedure can be modified to align more closely with known priors of the domain. The process of producing the model, and the model itself, can be defined more directly for the application. The specialization, however, must come with improvements over more general-purpose approaches to warrant the effort. Methodologies such as this are known as *informed machine learning*, in which prior knowledge is identified, a representation is chosen, and is integrated into the machine learning approach in some way. The full taxonomy of informed ML is presented in Figure 13. Informed ML is often used when data is scarce or when non-informed methods are simply under-performing, but they may be prioritized for many other reasons as well. Conventional ML strategies utilize data and set of processes to learn a solution. Informed ML however, integrates prior knowledge into the processes that learn a solution. As highlighted by [109], the taxonomy of informed ML can be outlined across three areas.

Source of Knowledge: A model developer must first consider *from where is the prior information produced?* Knowledge can be sourced from everyday experience, such as simple concepts that are considered well-known. For example, the knowledge that dogs have fur and they walk using all four limbs. Moving from this *world knowledge* to more explicit knowledge, as termed by [109], the solution must engage with deeper expertise. Subject matter experts in a particular business domain, for instance, might describe an underlying heuristic that they've utilized for their success. Social scientist, physicists, chemists, etc. may also contribute well studied formulations or theories that may help a model more easily discover worthwhile solutions.

Representation of Knowledge: Once a model developer is aware of key prior knowledge, they must ask - *how is the knowledge represented in the model building process?* This step of

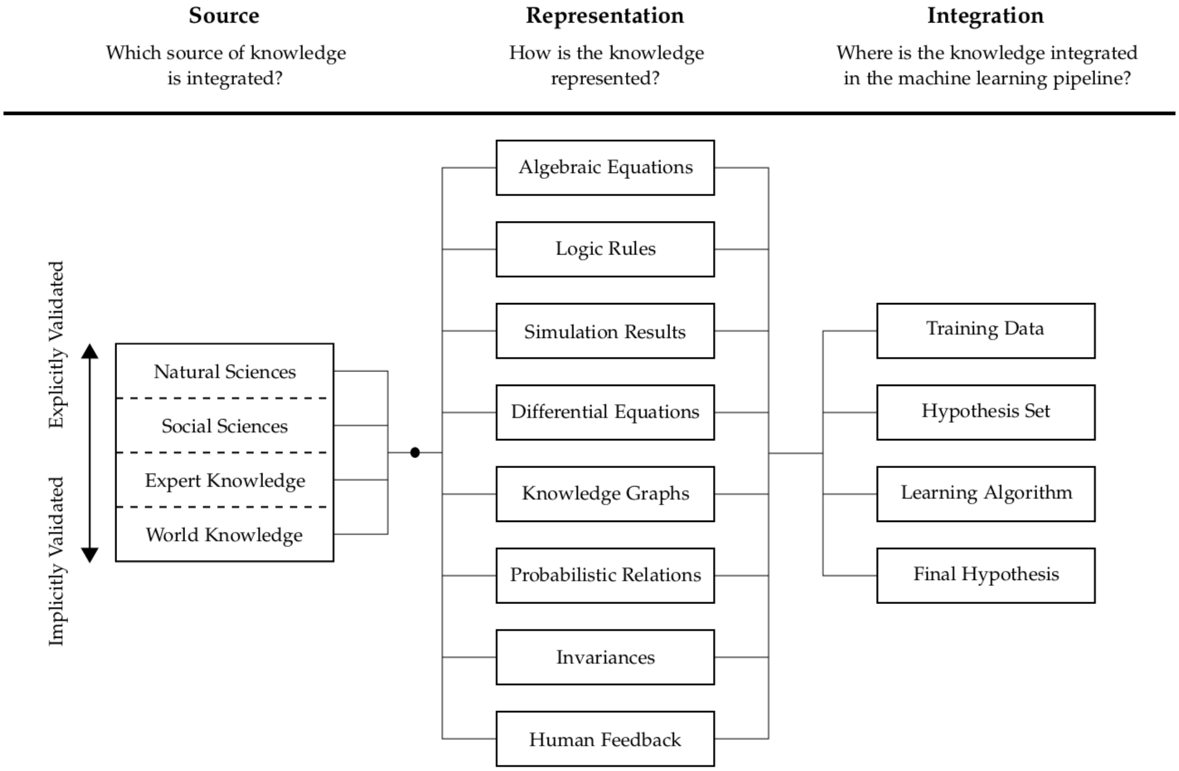


Fig. 13. Illustrations from [109]: Taxonomy of informed machine learning

informed learning is critical, and is one of the primary areas of research. The representation is the input to the integration stage, and is often implemented through concepts such as logic rules, invariances, relations, or even human feedback. As might be expected, the representation must realizable in some form, even if compromises must be made to represent the original information (e.g., rounding an irrational number or estimating a parameter).

Integration of Knowledge: With the prior knowledge clearly represented, a developer mush now consider *where in the ML process should the representation be integrated?* The survey in [109] highlighted four areas common for knowledge integration. The knowledge may be applied to the training data rather than the model's algorithm, it may be used to help define the solutions available to the model, it may directly influence the learning algorithm itself, or the knowledge can be used to validate the final output of a model. These techniques are not mutually exclusive, and multiple may be used to complement each other.

Informed ML has seen broad adoption in the physics domain, where experimental data can vary in size, but existing theories can help ML methods reach better solutions [110]. It's also well understood that domain knowledge integrated into a ML process can aid in the discovery of new insights [111]. In our contributions, we use informed ML in our design of Multi-SincNet, which is applied in both Chapter 4 and 3. Our approach sources knowledge from the natural sciences, represented as algebraic equations, in order to help reduce the scope of the hypothesis set.

2.4.2 Transfer Learning

Canonical ML strategies use learning algorithms to minimize a criterion given example data, often made up of input features and a desired target [1]. The resulting model can then be used to make predictions, or otherwise produce output, on new samples in the future. This strategy has limitations, however, when only a small portion of available training data has target labels for learning. Without sufficient data, learning and validating a useful model from only the labeled data can be challenging. In these scenarios, *semi-supervised* ML methods leverage a larger unlabeled dataset to improve performance on the smaller labeled dataset. However, semi-supervised learning must generally make the assumption that the distributions of the labeled and unlabeled data are the same. *Transfer learning* further generalizes this notion by allowing the domain, task, and feature distributions used in training and testing to be different. In transfer learning paradigms, methods attempt to improve downstream performance by adapting existing knowledge to new information [9]. Intuitive examples of how learning in related domains can enable fast adaption to new tasks are shown in Figure 14.

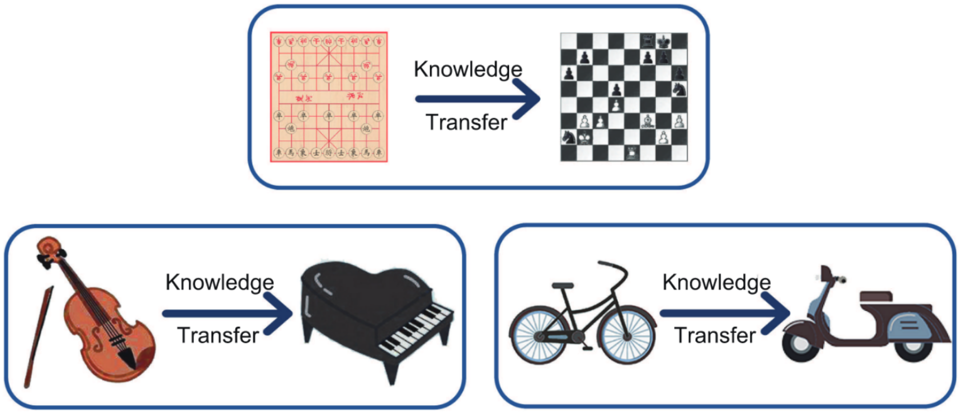


Fig. 14. Illustration from [112]: Intuitive examples of transfer learning

Transfer learning separates model development into *source* tasks and *target* tasks, and develops methods to adapt between them using various strategies [9] [112]. Methodologies establish an initial parameterization from a source task as an attempt to instill a model with additional information that will later improve performance during *fine-tuning* on a target task. The process of fine-tuning is often implemented as directly optimizing on the task the model is intended to solve, while leveraging the representations learned in the source task. An illustration of transfer learning, and how it relates to self-supervised and supervised learning, is shown in Figure 15.

In a *transductive* transfer learning approach, the source task and the target task are the same, but the domains vary. The feature space of transductive learning may differ between source and target task entirely, or simply the marginal distributions of the features may have changed. In contrast, *inductive* transfer learning refers to when the target and source tasks differ, regardless of the domain or feature space. It may be the case that abundant data exists for the source task, or perhaps more commonly, a majority of the relevant data is unlabeled [9]. Methods that learn from an unlabeled source task for later transfer to new tasks began as *self-taught* learning [114]. Today, *self-supervised learning* for DL is a popular method that does not require ground truth task labels. Instead, a task is derived from the nature of the data, one that requires the model to learn an informative mapping that is likely to translate to the future fine-tuning task [113].

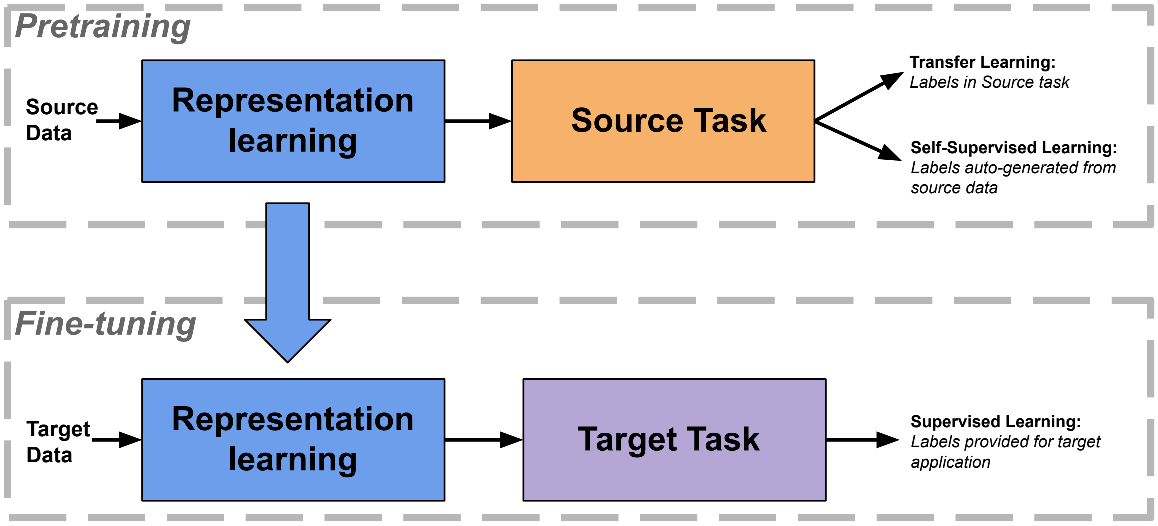

Pretraining
Source Data
Representation learning
Source Task
Transfer Learning: Labels in Source task
Self-Supervised Learning: Labels auto-generated from source data
Fine-tuning
Target Data
Representation learning
Target Task
Supervised Learning: Labels provided for target application

Fig. 15. Pretraining is a separate stage of model building and happens earlier in time to a fine-tuning procedure, in which the model is adjusted for the target data and task. [113]

2.4.3 Incremental Learning

In practice, AI and ML systems must contend with the *stability-plasticity dilemma* - the notion that an automated system must retain past knowledge while learning knew knowledge [115, 116]. If the system is too plastic, *catastrophic forgetting* may occur in which previously well-understood concepts are disrupted and can no longer be recalled [117, 115]. In a sense, the system has forgotten the knowledge it once held. The dilemma is considered a well-known constraint in neural systems, both biological and artificial, which hinders the ability to adapt to new data, tasks, or other aspects of the problem. Early work recognized a key dimension of this challenge as whether or not the training examples are presented *concurrently* or *sequentially* - that the issue primarily arises when new examples are presented separately and not together during a training session [117]. Intuitively, humans and other biological neural networks do not learn entirely concurrently. Instead, new concepts and tasks are introduced over time once previous tasks are sufficiently mastered.

Lifelong learning [118] and its more recent incarnation *continual learning* [116] are research areas that aim to address challenges of sequential learning in the face of stability-plasticity dilemma. The authors of [116] recently surveyed the field and describe it as research aimed at learning from an infinite stream of data. From this stream of data, new tasks may emerge (e.g. a new classification task), features may drift, and concepts relating features and outputs may evolve over time. In some more extreme cases, the *task boundary* or even the notion of the task may not be clearly delineated. Under these paradigms, the methods are motivated to discover the tasks or their mutation without supervision [119]. Importantly, the field recognizes that privacy is harmed when algorithms require access to past training examples during deployment Due to the required resources and potential privacy issues, the use of memory to store samples to combat catastrophic forgetting is considered a limitation to solutions. The authors of [116] describe three primary methods for continual learning in recent research.

(1) Replay Methods: As the name implies, these solutions either store raw samples or generate synthetic samples to include in the learning set for new tasks. A *rehearsal* based method use the samples as inputs to inform the training process, along with new task samples. A *psuedo-rehearsal* is when rehearsal is performed with synthetic samples or random samples. A *constrained optimization* approach uses stored samples to guide new model updates such that those updates minimally impact the previously learned task.

(2) Regularization methods: In order to avoid reliance on raw samples - preserving privacy and decreasing storage and memory usage - some approaches develop regularization methods to constrain new task learning. The constraints are intended to help stabilize the model using either small samples of data or an explicit prior. The *data-focused* methods are primarily *distillation* based approaches using the previous iteration of the model. *Prior-focused* methods estimate distributions of parameters and penalize the model for changing parameters measured as statistically important.

(3) Parameter Isolation: A more straightforward methodology in which new parameters are added to the model in order to accommodate new tasks. In practice, this can often be a new "classification head" that utilizes previously optimized features alongside other task-specific output layers. This prevents forgetting since the original parameters are never altered, but can significantly increase the model size if many unique output heads are needed.

In [119], the authors present Dual User-Adaption (DUA), a framework that separates server-side adaption from user-side adaption in a continual learning framework. The DUA is theoretically focused, aligning their formulation to steps in an algorithmic process and complements the conceptual framework we illustrate in Figure 1. Incremental learning has also received recent attention in work related to *federated learning* [120]. In federated learning, privacy preservation is approached by distributing the model optimization rather than centralizing the data storage for subsequent model optimization. Incremental learning was recognized as a more realistic paradigm for edge devices attempting federated learning in practice.

2.4.4 Measuring and Reducing Information Privacy Risk

In section 2.1 we discussed how AI systems can help bolster trust from users by prioritizing users' information privacy. In Section 2.3 we described DL for building AI through ML methods that requires data, or simulated environments, and produce new data in the form of learned parameters and extracted features. In Section 2.1.3 we described how a typical ML model, trained on potentially sensitive data like HAR recordings, is a representation of the data it was trained. The parameters of the models, or the models optimized form, are a potential attack vector for an adversary. This is important because in practice, releasing models for others to use requires either sharing the parameters (i.e., the model) or providing black-box access through an indirect interface. Either way, users have some level of access to the information represented by the model. Methods to understand how much information and what that information can reveal is the focus of this subsection.

A survey separated modern privacy research into ***data clustering*** and ***theoretical frameworks*** for managing privacy risks in shared databases [75]. Early methods proposed k-anonymity in which an *equivalence class* - group of values that can be used to aid in identifying the source of a record (e.g., gender, age, job, etc.) - must have at least k entries in the table in order to decrease the probability of re-identification [121]. The k-anonymity however does not ensure that *sensitive attributes* - values relating to the individual that the attacker is interested in knowing - are sufficiently diverse to prevent re-identification. The l-diversity method addresses this shortcoming by requiring that sensitive attributes be well-represented [122]. The l-diversity requirement can therefore be combined with a k-anonymity requirement for increased information privacy, but at the loss of information within the dataset. Further research developed t-closeness, which requires that each equivalence class's sensitive attributes be bounded with respect to the populations overall distribution [123].

Malicious individuals, or *attackers*, take advantage of information leakage within the model to discover aspects about the training data. Furthermore, how the model behaves - the distribution of its outputs for various inputs - may also hint at aspects of the training procedure or even the input samples themselves. When attempting to infer hidden information about a model's training dataset, one of the most fundamental questions that an attacker will want to answer is whether a given sample was even included in the model's training dataset. These attacks are known as membership inference attacks [25] and are not necessarily concerned with trying to re-identify a specific sample, only whether it was used in training the model. While the authors in [25] developed attacks against ML models for image analysis deployed as services, early work developed similar attacks for genomics databases [124, 125]. In order to measure the privacy risk of potential membership inference attacks, a number of strategies have been employed, including statistical tests and metrics use in ML for classification accuracy. Therefore, the measure of risk is effect size or the performance of the attacker's classifier [126].

A first step in information security is demonstrating the feasibility of an attack. Our work focuses on exploiting elements of a system to gain access to private information, but attacks may seek to destroy, weaken, or otherwise damage the target system. In Chapter 5, Section 5.3 we use two data-driven methods to evaluate the potential risk of information leakage for re-identification and membership inference. We accomplish this with experiments that simulate adversarial threat models in which an attacker uses ML techniques to either classify the individual's identity or their membership of a pretrained model.

CHAPTER 3

INTERPRETABLE FEATURE EXTRACTION FROM BIOELECTRIC NEURAL SIGNALS

3.1 Introduction

In Section 2.1, we discussed why a key aspect of trustworthy AI is *interpretability*, for both user understanding and expert validation. In this Chapter, we'll present two contributions to bioelectric sensor-based HAR using ML, both of which allow experts to interpret aspects of the model for their validation.

We begin in Section 3.2 with an expert-driven preprocessing pipeline and a grid search of ML hyperparameters for improved efficacy of Deep Brain Stimulation (DBS). The presented approach relies on EEG data, which exhibits two important issues related to HAR that were discussed in Section 2.2: distribution discrepancy between individuals and limited dataset sizes. While the feature extraction approach for predicting DBS treatment response has interpretable foundations, it cannot discover new features and requires developer guidance.

In Section 3.3, we use Multi-SincNet to learned interpretable parameters for speech detection from ECoG signals. Similar to the EEG data, the ECoG data in this work is limited and highly variable across participants. Our speech activity recognition approach discovers interpretable parameters, and compared to similar deep learning approaches, uses fewer parameters overall. These smaller personalized models aid in research, validation, and overall usability for end-users.

Both methodologies make informed assumptions about the underlying data generating process. We outline the informed components and how they fit into the taxonomy of [109] in Figure 16. In Section 3.2, prior literature and subject matter expertise guide the choice of features that a modeling process utilizes. In Section 3.3, the foundational natural science knowledge is integrated into the model, allowing the model to find similarly well-defined features as the experts.

3.2 Interpretable Preprocessing for Deep Brain Stimulation Efficacy

DBS has had success in treatment of movement disorders such as Parkinson's Disease (PD) [127, 128, 129, 130, 131, 132], Essential Tremor (ET) [133, 134, 135, 136], and dystonia [137][138]. The electrical stimulation of motor nuclei such as Globus Pallidus Interna (GPI), Subthalamic Nucleus (STN), and Ventral Intermedius Nucleus (VIM) of the thalamus generates an active volume around the electrode site modulating the neuronal tissue in that region. With therapeutic high frequency (125-185 Hz) stimulation, the relay of pathological signals through the sub-circuitries in the basal ganglia thalamocortical network is altered, ultimately alleviating the underlying motor symptoms. Previous work has shown a pathological coupling between phase and amplitude of EEG within the motor cortex [94][95]. Both DBS and dopaminergic medicines result in a return to a more normal EEG pattern [94, 95, 139, 140].

DBS treatment begins with surgery to implant leads in the brain region appropriate for the desired effect. The DBS leads have contacts over the distal 9mm of the implanted lead (Medtronic Inc.). These leads are connected to implanted generators which can be programmed at any time after surgery, altering contact location, amplitude, pulse width, and stimulation frequencies [141,

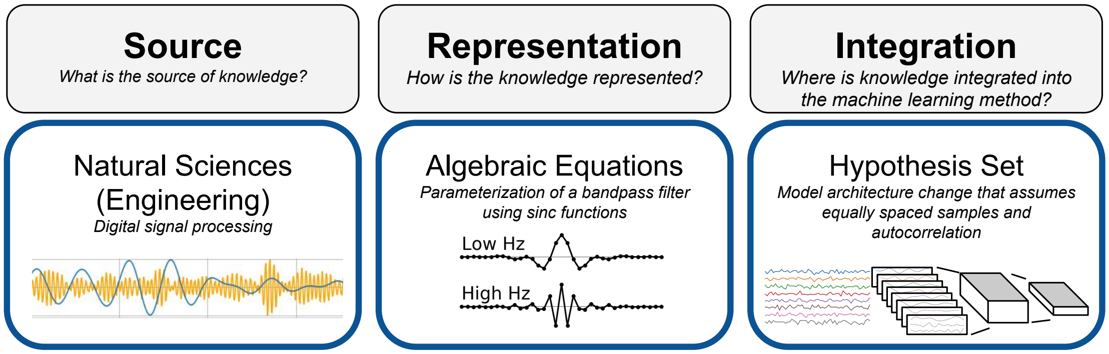

Source
What is the source of knowledge?
Representation
How is the knowledge represented?
Integration
Where is knowledge integrated into the machine learning method?
Natural Sciences (Engineering)
Digital signal processing
Algebraic Equations
Parameterization of a bandpass filter using sinc functions
Low Hz
High Hz
Hypothesis Set
Model architecture change that assumes equally spaced samples and autocorrelation

Fig. 16. The informed learning taxonomy membership for the work presented in this chapter. Methodologies from the digital signal processing domain are used to reduce learned parameters and create an explainable signal processing layer for deep learning.

142, 143]. After recovery from surgery, selection of the most effective parameters for symptom controls is required.

In these mapping sessions, the response to varied stimulation configurations is empirically evaluated for efficacy by monitoring patient response in real time. Side effects are simply brain responses other than the ones desired for efficacy. For instance, the contact in the ventral posterolateral (VPL) nucleus of the thalamus will elicit tingling in the contralateral hand or face, the contact in the VIM will stop the patient's tremor, and a contact within the internal capsule will elicit facial contraction [144]. All of these effects are routinely elicited during the mapping process.

While mapping and configuring the leads can often be straightforward, it is ultimately a visual-qualitative interpretation performed by an expert. The process can be hampered when application of the stimulation has delayed effects, such as DBS of the GPI region [145]. Occasionally, the patient will have side effects that cannot be easily categorized and can be related to anxiety or heightened awareness of internal stimuli [142] [143]. A comprehensive review of the complexities encountered during programming is given in [146]. It follows that an objective criteria for DBS efficacy could help reduce the configuration search space and improve the process.

For monitoring optimal response to DBS, EEG-based methods have advantages over other techniques, such as Positron Emission Tomography (PET), as the data can be acquired and analyzed repeatedly during a programming session as different parameters are selected and evaluated. Although PET scans have the ability to assess metabolic changes associated with DBS, this exposes the patient to radiation, and it cannot be used in an iterative process. Similarly the oxygen metabolic changes identified with Functional Magnetic Resonance Imaging (fMRI) are radically constrained by the Magnetic Resonance Imaging (MRI) environment and slower time course. Finally, the incorporation of EEG is relatively low cost.

In this section, machine learning based feature extraction and classification methods are applied to high resolution EEG data captured from 16 patients with DBS implants. Patients are fitted with a dense array EEG cap with 256 channels. EEG is recorded as the DBS mapping procedure cycles through stimulus configurations. For each patient, EEG data is also captured without DBS being applied. The resulting dataset is annotated for the location of the lead within the basal ganglia/thalamus and the clinical efficacy of the DBS parameters.

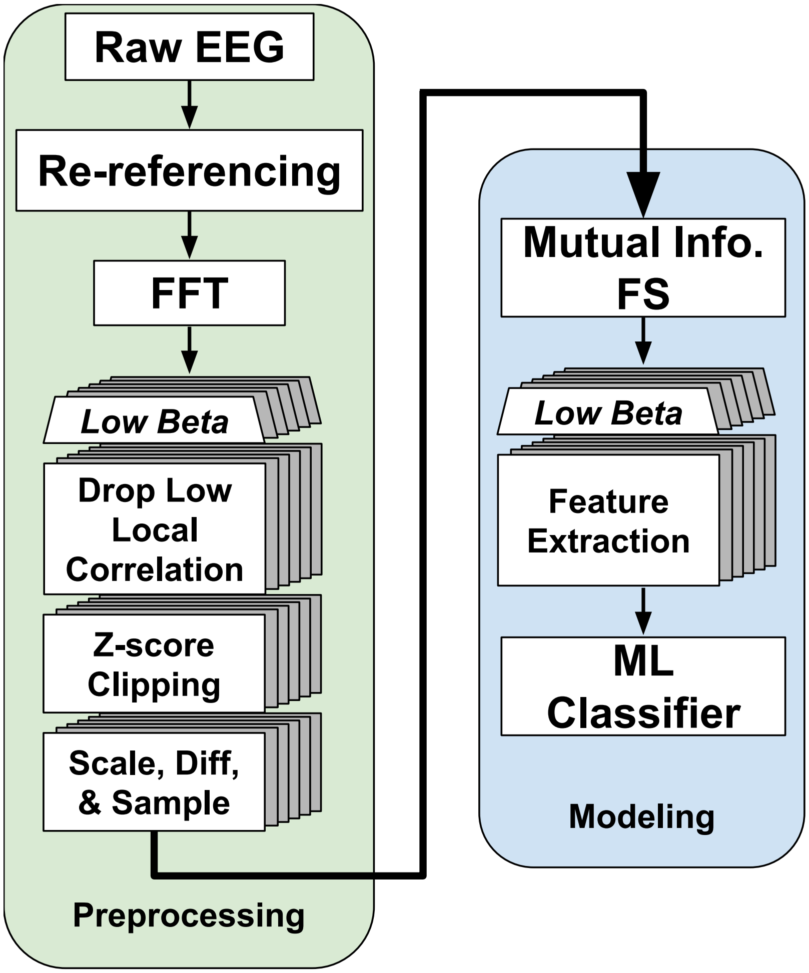


Fig. 17. Preprocssing pipeline for classification of Deep Brain Stimulation - design devisions are driven by domain and task -oriented restrictions and expert knowledge.

With this dataset, we explore the informative capacity of EEG to assist in DBS treatment. Specific problems include whether DBS is being optimally applied and what aspects of brain activity are affected during DBS. We consider these challenges through two separate classification tasks: (1) Detecting active stimulation and (2) classifying the region of the brain undergoing stimulation. We compare resulting performance and features selected, split across three stimulation regions and 16 patients. Our results demonstrate the clear potential for reliable classification within these tasks, with both detection of DBS across patients and DBS region discrimination consistently achieving precision over 0.6 while still maintaining useful recall.

A total of 16 patients participated in the data collection procedure, each with stimulus in either the GPI, VIM, or STN regions of the brain. There are 8 patients with GPI DBS (6 PD, 1 dystonia, 1 Tourettes), 6 patients with VIM DBS (2 PD, 4 ET), 1 patient with STN DBS (PD), and 1 PD patient with VIM on the right and STN on the left.

EEG data acquisition is performed with an Electrical Geodesisc Inc Dense Array system containing 256 sensors, sampled at 1KHz using the EGI GTEN 100 Amplifier via the EGI NetStation 5 software. For optimal contact each patient's head circumference is measured in order to select the most appropriately sized net of sensors. A 2D projection of electrode placements

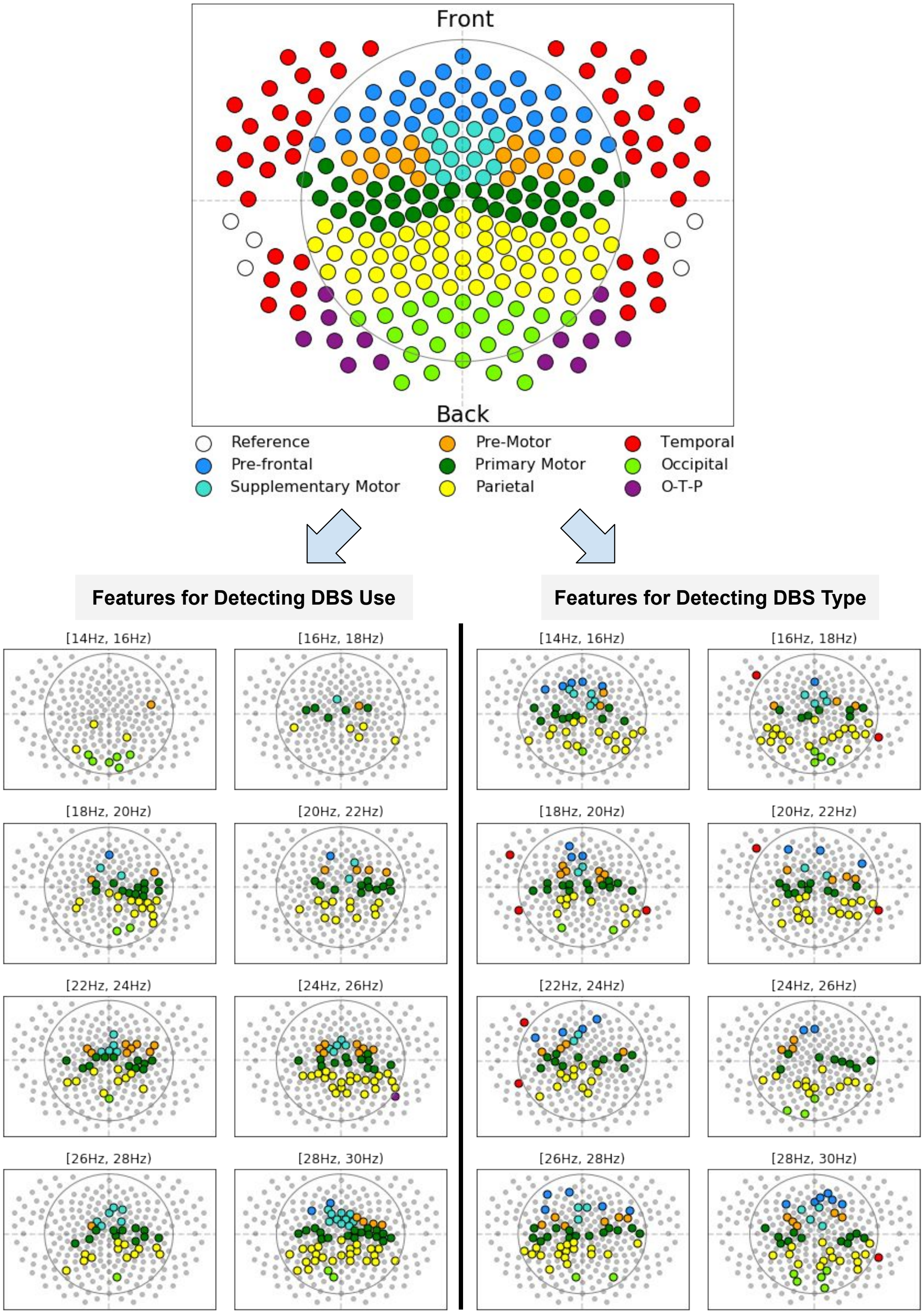


Fig. 18. Available sensors, their corresponding brain regions, and the features utilized by the best performing models. The selected regions vary by task, but greatly reduce the feature size of the downstream grid search.

Table 2. Summary of Deep Brain Stimulation Patients

ID	Stim. Region	DBS Type	On/Off Length	Partition
1	GPI	Bilateral	02:16 / 01:48	Train
2	STN+VIM	Bilateral	02:00 / 01:38	Train
3	VIM	Left	02:04 / 01:52	Holdout
4	GPI	Bilateral	04:46 / 04:22	Holdout
5	GPI	Bilateral	02:50 / 02:30	Train
6	GPI	Bilateral	01:56 / 02:08	Train
7	STN	Bilateral	02:00 / 01:58	Holdout
8	VIM	Bilateral	02:22 / 01:42	Train
9	VIM	Bilateral	03:54 / 03:46	Train
10	GPI	Bilateral	02:12 / 02:22	Holdout
11	VIM	Bilateral	03:44 / 04:44	Holdout
12	GPI	Bilateral	03:54 / 02:20	Train
13	VIM	Bilateral	01:32 / 09:14	Train
14	VIM	Left	04:02 / 04:14	Holdout
15	GPI	Bilateral	01:26 / 02:54	Train
16	GPI	Bilateral	05:04 / 01:26	Holdout

and brain regions are illustrated at the top of Figure 18.

EEG data is first recorded while the technician monitors for artifacts and adjusts sensors in real time to address issues. Next, DBS is turned off and EEG is captured for 2-5 minutes or as long as tolerable to the patient. With the baseline collected, the electrode mapping begins in a standard fashion while EEG data is collected. Once an optimal response is achieved, another 2-5 minutes of data is recorded.

After initial data collection, the recording session is segmented into several smaller datasets. These include segments of optimal DBS response resulting from successful mapping, poor DBS response encountered during mapping, and periods when DBS is switched off. In this work, data segments captured when DBS is off are referred to as *DBS-OFF* and data captured during optimal DBS are referred to as *DBS-ON*.

3.2.1 Feature Extraction and Modeling Pipeline

The data undergoes several processing stages before being passed to a supervised classifier. These steps are categorized as part of either the preprocessing pipeline or the model pipeline, as illustrated in Figure 17. Section 3.2.1 details the unsupervised preprocessing and Section 3.2.1 describes target-oriented feature selection, extraction, and modeling.

Before processing, patient data are split into a train partition and a holdout partition. The partitioning is stratified across the stimulation region to ensure variations arising from the region are well-represented in both sets of samples. See Table 2 for a summary of each patient's stimulation region, type, experiment length (MM:SS), and partition. The train set is used to tune the overall pipeline, while the holdout set is reserved for the evaluation of the best performing processes developed on the train set.

Preprocessing Pipeline

The raw EEG voltages are first re-referenced to the average voltage of the sensors nearest the mastoids, since these sensors tend to receive reduced signal from the brain. Specifically, for each sample of 256 real-valued potentials, the sensors nearest the back of the ear are averaged and the resulting mean is subtracted from all sensors, including the reference sensors. The channels used as references are identified in Figure 18 as white sensors.

Next, the Fast Fourier Transform (FFT) is applied to the re-referenced EEG magnitudes. The FFT is applied in 1 second sliding rectangular windows with a step size of 1 second. From each channel, we extract the average response from 8 bands characterized by a center frequency f_0 and a bandwidth of 2Hz, yielding the region $[f_0 - 1, f_0 + 1)$. Center frequencies begin at 15Hz and continue in steps of 2Hz, ending with the inclusion of $f_0 = 29$Hz. The FFT's windowing procedure and the band extraction reduces the number of samples and increases the number of features, making conservative anomaly detection and feature selection critical for reliable results.

Next, anomaly detection must be applied in order to remove artifacts arising from physiological differences or irregularities in electrode connectivity. The first step of automatic anomaly detection is applied to the bands retrieved from the FFT process. We examine spatially local correlation to identify sensors that behave poorly over time. This is approached by calculating the Pearson correlation coefficient for the r_n nearest sensors to each sensor. Distance is measured on the 2D plane shown in Figure 18 using Euclidean distance. Any sensor with a median neighbor correlation less than a specified positive outlier threshold, r_T, is considered anomalous. This rule is derived from the spatial locality of sensors, which generally results in a strong correlation over time between neighboring sensors. We select $r_n = 7$ and then calculate $r_T = 0.72$ as the 90th

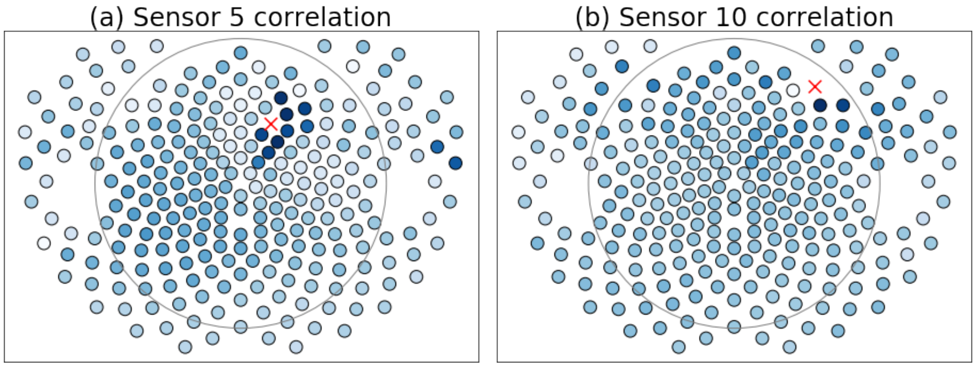


Fig. 19. Example of correlation-based outlier detection using patient 5's Deep Brain Stimulation (DBS)-OFF beta band. (a) Sensor 5 is highly correlated to most of its neighbors, while (b) sensor 10 is irregularly not correlated with its neighboring sensors. Counter-intuitive to common linear statistical modeling, uncorrelated sensors are considered anomalous.

percentile of median neighbor correlation for all sensors across all experiments in the training set.

Sensors failing the local correlation threshold in the train set are dropped from the remainder of the processing for that band in all datasets. We choose this treatment, rather than imputation, to leverage the redundancy of the 256 channel EEG and to avoid introducing unwanted bias from imputation. However, since only the training partition is used to identify poorly correlated sensors, new correlation anomalies in the holdout must be imputed. Therefore, sensors identified as anomalous through local correlation in the holdout are replaced with the median value from their r_n neighboring sensors at each sample. This is performed before dropping sensors in the holdout that were identified in the train set.

After addressing sensors that behave poorly over time using local correlation, we examine each sensor value with respect to the distribution of values captured by a sensor in an experiment. We use the modified Z-score, defined in Equation 3.1, to identify anomalous sensor values. The modified Z-score uses the median instead of the mean in its calculation, and is therefore well-suited for identifying outliers [147].

$$Z_{s_i} = \frac{x_{s_i} - median(x_s)}{1.4826 \times \mathrm{MAD_s}} \tag{3.1}$$

$$\mathrm{MAD_s} = median(|x_s - median(x_s)|) \tag{3.2}$$

Where x_s is all data for sensor s and x_{s_i} is the ith value for sensor s. We select $Z_{s_i} > 5$ as our threshold for outliers, such that each value outside this threshold is clipped to the range $[-Z_s = 5, Z_s = 5]$. Given equations 3.1 and 3.2, a unique threshold value, X_T, exists for each sensor across each experiment and is calculated using Equation 3.3:

$$X_T = 1.4826 \times 5 \times MAD_s + median(x_s) \tag{3.3}$$

Any value in a sensor with $|Z_s| > 5$ is replaced with the sensor's X_T. While this artificial ceiling will still carry some portion of its original information, it no longer has as large an influence on down-stream processing.

After clipping, each experiments' band-level datasets are scaled to the interval [0, 1] by dividing by each channel's respective maximums. A delta transform is then applied, taking the sample-to-sample difference through time. This centers the values around zero and helps remove any level shifts that may leak experiment information, without involving additional parameters. Finally, to balance the dataset and ensure fair representation, 140 samples are drawn (with replacement) from each band-level dataset for every experiment. This sampling procedure is only performed on the training set.

Modeling Pipeline

Following the preprocessing described in Section 3.2.1, the bands are recombined for univariate feature selection using Mutual Information (MI). For every feature, MI between the feature and the target is computed. The features are ranked by their MI score and the top 25% of features are kept. These features are then separated back into their respective bands before being passed to a model.

Each model is the combination of feature extractions at the band level concatenated and passed to a supervised classifier. We use the SciKit-Learn machine learning library [148] for feature extraction, modeling, and searching the hyperparameter space. Four feature extraction techniques are compared, including three unsupervised methods: Principle Component Analysis (PCA), Independent Component Analysis (ICA), and Agglomerative Feature Clustering (AFC) with mean pooling. Common Spatial Patterns (CSP) [149], a supervised technique popular in Quantitative Electroencepahlography (QEEG), is also included. Extracted features are passed to one of five supervised classification models. We compare Logistic Regression (LR) with L2 regularization, classification from K-Nearest Neighbors (KNN), random and gradient boosted ensembles of trees (Random Forest (RF) and Gradient Boosting (GB)), and Radial Basis Function (RBF) Support Vector Machine (SVM).

The four feature extraction methods and five model types result in 20 distinct extraction+classifier models to be examined and tuned. Due to the size of the search space, a randomized grid search is used to explore potential hyperparameters using K-Fold cross-validation on the training patients. Models are ranked and selected by their precision score. The folds are grouped at the patient level such that no patient's samples are included in both the holdout fold and the training folds. The K-fold strategy varies between the modeling task, so further detail is given in Section 3.2.2. Feature extraction methodologies are not mixed in a single model, meaning only one of PCA, ICA, CSP, or AFC is applied in a given model, but each band's extraction procedure has independent hyperparameters.

3.2.2 Results

Towards guided DBS for improved efficacy, we examine ML methodologies on two binary classification problems: detecting DBS and discriminating between regions of DBS across patients. Both problems are approached using the preprocessing and modeling pipelines described in Sections 3.2.1. Once the best hyperparameters for each of the 20 extraction+classifier models are selected, we then select the best extraction method for each classifier based on cross-validation performance. These 5 resulting extraction+classifier models are then examined on the holdout partition to evaluate generalization performance.

Table 3. Stimulation detection best CV precision scores

	CSP	ICA	FC	PCA
GB	**0.83**	0.81	0.79	0.82
KN	0.70	0.68	0.68	0.69
LR	0.52	0.52	0.53	0.52
RF	0.82	**0.82**	**0.80**	0.82
SVM	0.75	0.57	0.83	**0.83**

Table 4. Stimulation detection model performance on holdout patients

	N Extracted	**Accuracy**	**F1**	**Precision**	**Recall**
PCA - SVM	34	0.56	0.40	**0.73**	0.28
PCA - RF	36	**0.62**	**0.59**	0.70	0.51
AFC - LR	32	0.49	0.51	0.52	0.51
CSP - KNN	28	0.54	0.55	0.58	**0.52**
CSP - GB	48	0.53	0.45	0.61	0.36

Detecting DBS Across Patients

We first examine the ability to detect active DBS in a patient, regardless of DBS type. To accomplish this, the training cohort and their DBS-ON and DBS-OFF segments are combined into a single set of training samples. Samples taken from DBS-ON belong to the positive class, while DBS-OFF samples are assigned to the negative class. The resulting holdout dataset has a 0.51 target rate across 2,584 samples. The features selected, through anomaly treatment and then feature selection, are depicted in Figure 18. A 9-Fold, leave-one-patient-out cross-validation scheme is used in this experiment in order to encourage cross-patient generalization.

The best cross-validation results of the hyperparameter search are given in Table 3. The holdout results for the top performing classifier+extration pairs are given in Table 4, alongside the total number of features extracted across bands. We find that PCA-SVM and PCA-RF are most successful regarding precision on the holdout set, but the SVM classifier only recalls less than a third of the positive samples, leaving PCA-RF with the highest F1 score. The CSP-AFC model, a high bias estimator coupled with supervised extraction, achieves the best recall while still maintaining a competitive F1 score. We take the best performing model on the cross-validation set and produce a learning curve, shown in Figure 20-A, which appears to asymptote, suggesting the need for more informative features.

Table 5. Stimulation region classification best CV precision scores

	CSP	ICA	AFC	PCA
GB	0.52	0.60	0.60	0.65
KNN	0.52	0.58	0.55	0.52
LR	0.53	0.53	0.52	0.53
RF	**0.59**	**0.63**	**0.61**	**0.65**
SVM	0.57	0.51	0.53	0.52

Table 6. Stimulation region classification model performance on holdout patients

	N Extracted	**Accuracy**	**F1**	**Precision**	**Recall**
CSP - SVM	48	**0.66**	0.64	0.68	0.61
PCA - RF	36	0.60	0.58	0.62	0.55
PCA - LR	30	0.54	**0.65**	0.53	**0.85**
ICA - KNN	26	0.59	**0.65**	0.58	0.74
PCA - GB	44	0.65	0.62	**0.68**	0.58

Classifying DBS-ON Stimulation Region

Next, the ability for ML classifiers to separate active DBS-ON region is examined. For this experiment, only DBS-ON samples are utilized. Furthermore, while three separate stimulation regions are present in our dataset, STN is poorly represented with only two patients receiving this type of treatment, with one of these patients receiving both VIM and STN stimulation. For this reason, we combine VIM and STN treatments into a single class and contrast these samples against GPI treatments. Thus, we examine a binary classification task with GPI treatments assigned to the positive class and VIM+STN assigned to the negative class. The resulting holdout dataset has a 0.53 target rate in its 1,379 samples. The features selected as a result of anomaly detection and feature selection are illustrated in Figure 18.

To match our DBS detection experiments, we again apply 9-fold cross-validation. However, because each patient only contributes to one class in this dataset, at least two patients must be used in the holdout folds. Thus, for each of the 9 folds, one patient from each class is randomly selected into the holdout fold. Each patient is included in the holdout at least once. This scheme supports a higher number of unique folds with only a fraction of the underlying groups.

The best cross-validation precision scores for each of the 20 models examined are given in Table 5. The holdout results for these models, along with the total count of extracted features, are given in Table 6. In contrast to the DBS detection models, we find that random forest based models achieve the best precision, regardless of extraction strategy. Furthermore, the random

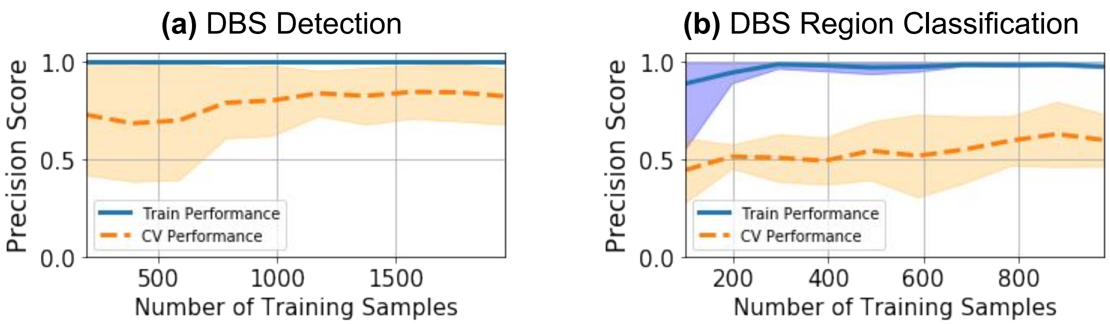


Fig. 20. **(a)** Learning curve with 9-fold grouped cross-validation for DBS detection using CSP-GB model. A subtle upward trend in the validation data that levels off suggests the need for more meaningful features and better regularization. **(b)** Learning curve with 9-fold grouped cross-validation for DBS-ON stimulation region classification using Principle Component Analysis (PCA)-RF model. The model validation scores trend up, suggesting the utility of additional data. However, high variance across folds show the difficulty in generalizing across patients.

forest model maintains its performance from the cross-validation set to the holdout, while other low variance models managed to surpass it in holdout performance. The inconsistency between cross-validation scores and holdout scores in this classification tasks suggests a need for better regularization and additional data. This is further explored by producing a learning curve for PCA-**AF!** (**AF!**) model, shown in Figure 20-B. The learning curve results show a clear upward trend in the validation scores, confirming the need for more data in order to avoid spurious correlations.

3.2.3 Related Work

Early efforts to decode the EEG response of the brain to DBS concentrated on the peak amplitude and latencies of the Evoked Potential (EP) in the area of the motor cortex. Work in [136][150][151] provides insight into similarities and differences between the various locations of the implanted stimulator. In contrast to the extensive study of EPs to STN stimulation, much less has been explored with GPI EPs, with existing work focusing on dystonia patients [152][153].

Source localization algorithms have been useful in identifying the affected EP in both the anatomic and time domains. Laxton et al mapped the brain areas that are affected by electrical stimulation of the fornix in AD patients [154][155], demonstrating activation of the ipsilateral hippocampal formation and the medial temporal lobe. This data was consistent with the PET data in the same patients. The technique has also been used in mapping the response to Brodman area Cg 25 for the treatment of depression [156].

Beta oscillations have long been recognized as the idling rhythm of the motor cortex. The discovery of the beta band in the STN region of Parkinsonian patients brought renewed focus to this unique oscillation and its potential role in PD pathophysiology. De Hemptine et al [140] has demonstrated that STN DBS reduces this excessive phase amplitude coupling seen in PD. Although the original experiments were conducted intra-operatively using electrocorticography, a similar finding has been demonstrated noninvasively with EEG while analyzing the effect of medications on the excessive Phase Amplitude Coupling (PAC)[94][95]. This study demonstrated increased PAC off medications as compared to on medication as well as controls. This work suggests that the EEG signature of effective DBS stimulation may be disease specific rather than nucleus specific, with effective stimulation resulting in alteration of the abnormal oscillatory characteristics of the patient's disease state.

In general, the work in this section continues the trend of applied QEEG techniques that have been popular for brain computer interfacing, including motor imagery [157, 158, 159], spike detection [160][161], and transcranial stimulation [162]. Finally, while this work focuses on EEG data, other inputs for assessing effectiveness, such as video monitoring [163][164], may prove valuable in future work.

3.2.4 Discussion

We approach the problem of DBS classification using an array of common feature extraction techniques and machine learning models. Results clearly demonstrate successful detection of DBS, as well as classification of DBS region. We find that an SVM applied to features extracted using PCA to be the most precise when detecting DBS across patients. A decision tree-based gradient boosting ensemble, paired with PCA, achieves the highest precision for identifying DBS region. Overall, the majority of models beat the baseline precision, lending support for future effort in this approach to improving DBS efficacy.

Several areas of our work could be expanded or considered more closely in future work. First, a broader range of feature extraction techniques should be considered, especially those that account for the spatial context of the sensors. This includes how regions may couple in the time and frequency domain, as shown to be relevant in prior work. Additionally, our anomaly treatment strategy may benefit from domain knowledge. For instance, sensor correlation checks may be more sensibly performed with neighbors from within the same brain region only. Other areas of improvement include stronger regularization for both complex models and supervised feature extraction [165].

3.3 Interpretable Speech Detection from Brain-Computer Interfaces

BCI hold the potential for a direct connection to thoughts and intentions, as well as direct neural control of external devices [166]. Due to superior spatial resolution and spectral bandwidth, invasive BCI's have advantages over non-invasive BCIs for more intricate direct neural control applications. ECoG is an invasive measurement of the electrical potentials generated from the neocortex of the brain [167]. ECoG signals have been shown to successfully control the movement of an upper-limb neuroprosthetic [168], typing interfaces [169], as well as decoding speech processes [170].

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

We present a component model, SincIEEG, based on a CNN architecture developed for the task of speech activity detection [171]. The model is designed as a gateway, constantly monitoring brain activity to identify the segments pertinent to speech production. These detected segments can then be sent to downstream models for subsequent speech decoding and synthesis. SincIEEG, unlike a traditional CNN, learns a set of bandpass filter coefficients at its input layer. This provides several advantages over a traditional CNN since the number of required model parameters is significantly reduced by comparison, making it computationally efficient in terms of training and implementation. This compactness allows for flexibility without increasing the optimization problem. Moreover, unlike most traditional CNNs, the SincIEEG model has the distinct advantage of yielding interpretable parameters. The bandpass filters learned by SincIEEG can be visualized and equated to conventional spectral brain features.

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

The SincIEEG is a Multi-SincNet based convolutional deep learning architecture adapted for real-time detection of human speech from ECoG input signals. Originally presented for hand-pose classification from myoelectric sensor readings in the next chapter's contribution [172], and based off the work in [171], the Multi-SincNet architecture learns the coefficients for a set of parallel Finite Impulse Response (FIR) bandpass filters, applied across the input channels. Subsequent convolutional layers learn kernels that aggregate across time and bandpass frequency dimensions. A final global view, established by a fully connected layer and sigmoid activation, classifies either 'speaking' or 'not-speaking' from labeled data. Figure 21 illustrates the SincIEEG model and its layer configurations.

In overview, the inputs to the model are 500 ms windows of raw Intracranial Electroencephalography (IEEG) data (300 time samples) with a stride of 2 ms (1 time sample). The corresponding label was whether the participant was speaking on the final time sample of the window. Each 500 ms window represents one training sample for the model. A model was trained for each participant, using all of the quality electrodes available. Electrodes over the auditory cortex were excluded for a model validation check, detailed in the following section. A K-fold training methodology was used and is detailed further in Section 3.3.3.

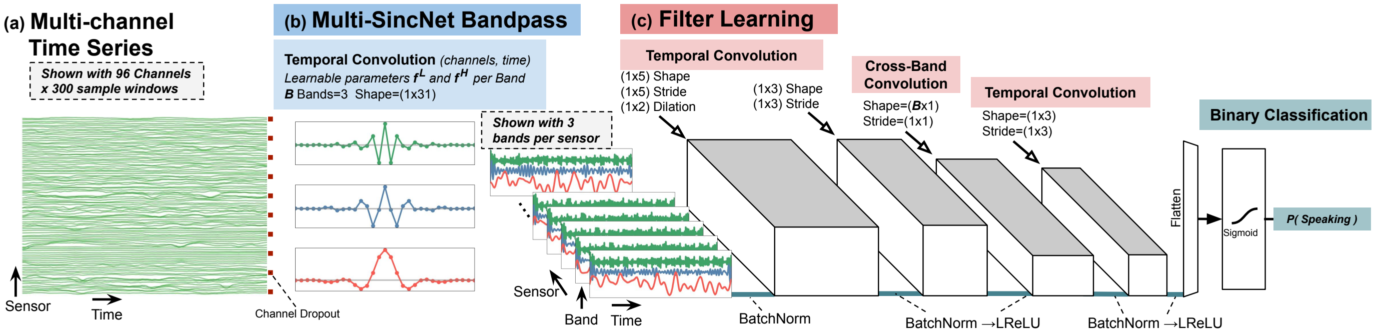


Fig. 21. The **SincIEEG** deep learning architecture: a classification model composed of a Multi-SincNet input layer and multiple subsequent convolutional layers. **(a)** SincIEEG takes raw multi-channel ECoG time series data as input, with channel dropout for improved regularization. **(b)** Multi-SincNet learns bandpass filter parameters to decompose the input signal - illustrated here with three pass-bands. **(c)** The filtered signals are normalized with respect to the band dimension using spatial normalization before convolutional layers learn kernels across time and pass-bands. All hidden layers use batch normalization for regularization and *Leaky Rectified Linear Units* for activation. The model predicts the likelihood of speaking using a *Sigmoid* activation at its output layer.

This architecture, was developed and implemented using Pytorch [173] deep learning Python library. Other critical software libraries used for development and discovery include matplotlib [174], numpy [175], pandas [176] [177], seaborn [178], SciPy [179].

3.3.1 SincIEEG Architecture

The first layer in the SincIEEG model is a Multi-SincNet layer, an extension to the the Kaldi speech framework's [180] SincNet, which applies a SincNet to each of the incoming sensor channels. The SincNet and Multi-SincNet layers are discussed in more detail in Chapter 4, but we briefly overview the method for this contribution. A SincNet layer learns a configurable number of bandpass filters, parameterized through two cutoff frequencies, f_L and f_H. The Multi-SincNet layer can therefore be used to decompose a collection input signals into a fixed set of learned bands.

In equations 3.4 and 3.5, multiple filters are conceptualized as vectors of low and high cutoffs, F_L and F_H respectively, identifying regions of the input's spectrum that the model uses for classification. These vectors are a parameterization of a SincNet layer, which is shared in our experiments across all sensors $s \in S$.

$$F_L = \{f_L^0, f_L^1, ..., f_L^{i=B-1}\} \in \mathbb{R}^+ \tag{3.4}$$

$$F_H = \{f_H^0, f_H^1, ..., f_H^{i=B-1}\} \in \mathbb{R}^+ \tag{3.5}$$

$$K : (f_L, f_H, f_s) \mapsto \mathbb{R}^W \tag{3.6}$$

$$\text{SincNet}(F_L, F_H) = \{K(F_L(i), F_H(i))\} \tag{3.7}$$

$$\text{Multi-SincNet} = \text{SincNet}_{\mathrm{F_L,F_H}}(s) \ \ s \in S \tag{3.8}$$

Sharing bandpass filters across each sensor reduces parameters, improves model latency, and regularizes the treatment of sensor data. Each FIR filter, k is implemented as a set of kernel coefficients and applied through convolution with the input signal X.

$$X \otimes k_{(f_L, f_H)} = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} X[i] * k_{(f_L, f_H)}[j-i] \tag{3.9}$$

Where X is the input signal and k_{f_L, f_H} is the vector of kernel coefficients that allows frequencies in $[f_L, f_H]$ to remain in the signal. Additional details on the calculation of k coefficients and how they compare to learned kernels can be found in [171].

Filters are initialized to uniformly sub-divide the majority of the available spectrum (i.e., 0-300 Hz) with a 3 Hz region of overlap between adjacent bands. The original Kaldi implementation initializes bands starting at a low-cutoff of 30 Hz, but we reduce this minimum starting frequency to 10 Hz to help encourage use of lower frequencies that may be relevant for this application [181]. The Kaldi SincNet implementation also includes a minimum frequency and minimum bandwidth constraint, which we configure to be 1 Hz and 3 Hz, respectively. Kaldi enforces these minimums by increasing the absolute value of the learned low-cutoffs and bandwidths by their respective minimums. Future work should explore the impact of different potential initialization schemes.

Activation

ReLU, defined as $y = max(0, x)$, provide a linear gradient for all input $x \in \mathbb{R}^+$ and 0 gradient for $x \leq 0$. With zero-centered bandpass outputs, a large portion of values will not have a gradient with ReLU activation. Instead, the Leaky Rectified Linear Units (LReLU) provides a small gradient for $x \leq 0$, while still being non-linear and computationally simple. The LReLU activation is defined in equation 3.10, where we use the default $\alpha = 0.01$ for all our experiments.

$$\textit{Leaky ReLU}(x) = \max(0, x) + \alpha * \min(0, x) \tag{3.10}$$

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

Batch Normalization

The amplitude of the output from the Multi-SincNet filters scale directly with the amplitude of the input signal. Between-sensor relative magnitudes are important to maintain, so we avoid scaling at the sensor dimension of intermediate data in the early layers. Brain dynamics are not evenly distributed in the frequency domain, however, and will tend to have higher amplitudes at lower frequencies. This means the additional bandpass dimensions may be distributed at different scales, making it difficult to learn shared kernels in subsequent convolution layers. Furthermore, the scale of the intermediate values may shift as the cutoff frequencies of the learned bandpass filters are optimized.

Therefore, in order to balance influence when learning kernels applied across bands, and to scale hidden outputs to activation regions, a spatial batch normalization [182] is applied at the band dimension in the three hidden outputs following the Multi-SincNet input layer. Re-scaling each band independently maintains within-band relative dynamics that can be learned using shared weights.

$$\mu_f = \frac{1}{BST} \sum_{b=0}^{B-1} \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} X[b, s, f, t] \tag{3.11}$$

$$\sigma_f = \frac{1}{BST} \sum_{b=0}^{B-1} \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} (X[b, s, f, t] - \mu_f)^2 \tag{3.12}$$

$$y = \frac{X - \mu_f}{\sqrt{\sigma_f + \epsilon}} * \gamma + \beta \tag{3.13}$$

$$for f \in F$$

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

Monte Carlo Dropout

Sensor systems with many highly responsive input channels may have spurious errors or drift, and sometimes must be removed in pre-processing. Additionally, for general tasks such as speech activity detection from an ECoG array, some important brain regions may have multiple sensors covering them, resulting in high co-linearity across channels. To regularize co-linearity across sensors, channel dropout [183] is applied on the input to the model during training. Channel dropout on the sensors zeros all signal values for a sensor with an independent Bernoulli random number parameterized by probability p. It is common to avoid using dropout when using batch normalization since the noise caused by the dropout will skew the mean and variance statistics used in normalization towards zero. However, for SincIEEG's, the data modality is already centered at zero, and the practical application motivates robustness to sensor dropout.

3.3.2 Data Collection

Participants

ECoG data were recorded from 5 participants with pharmacoresistant epilepsy undergoing clinical monitoring for surgical planning. No participants reported hearing deficits. In all cases, a tumor was not the source for the seizures and no lesions were indicated by any electrode used for

Table 7. Electrodes by Participant

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

analysis. All participants gave written informed consent and the study protocol was approved by the institutional review boards of Virginia Commonwealth University; University of California, San Diego; Old Dominion University; and Mayo Clinic, Florida.

Participants were implanted with subdural electrode grids or strips (Ad-Tech Medical Instrument Corporation, 1-cm spacing) based purely on their clinical need. Electrode locations were verified by co-registering preoperative MRI and postoperative computerized tomography scans. For combined visualization, electrode locations were projected to common Talairach space. Electrode locations were rendered using NeuralAct [184], as shown in Figure 34. While brain areas associated with speech are predominantly found on the dominant hemisphere, which is the left hemisphere in the majority of right-hand dominant people, the neural correlates of speech production are not exclusively localized in the left hemisphere [185, 186]. For this reason, both left and right hemisphere cases are evaluated. In total, ECoG activity was recorded from 416 (96 left hemisphere, 320 right hemisphere) subdural electrodes. Of these, electrodes that exhibited unnatural signal anomalies based on visual inspection were excluded from the analysis, leaving 364 electrodes (96 left hemisphere, 268 right hemisphere). For each participant, the number of electrodes implanted, analyzed, and identified as not located over the auditory cortex (non-auditory) are provided in Table 7.

Task

Participants were instructed to read aloud single words presented in sequence on a computer screen while their brain activity and voice were simultaneously recorded. The words were selected from a bank of 431 unique words, split into 4 sets of 115-116 words. The bank of words are primarily monosyllabic and comprised of the Modified Rhyme Test [187], supplemented with additional words to better reflect the phoneme distribution of American English [188]. While this experimental paradigm was originally designed to examine neural correlates of American English phonemes [189], the data are being used in the present analysis exclusively for speech activity detection without consideration of phonetic aspects.

The experiment begins with a fixation cross at the center of the screen. The cross is then replaced by a word that stays on the screen for 2.5 seconds. The word is then replaced with the cross for 0.5 seconds, before the next word is presented. Words are chosen randomly from the set of 115 words for each session and each session contained different subset of words. Participants completed between 2 and 4 sessions, depending on willingness and ability to complete the sessions.

Data Acquisition

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

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

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

Audio data were recorded in parallel using a Blue Microphones Snowball iCE USB microphone connected to the research computer, sampled at 48 kHz. All data recording and stimulus presentation were facilitated by BCI2000 software [190].

Speech Labeling

Speech labels used for training the model were made in reference to the stimulus cue of the word being presented in the experiment. Every time-sample from 0.5 seconds after the word presentation cue to 1.5 seconds after the cue were labeled as ‘speaking’. Every time-sample from 2.0 seconds after the word presentation cue to 3.0 seconds after the cue were labeled as ‘not speaking’. The other segments, from the cue to 0.5 seconds after, and from 1.5 to 2.0 seconds after, were purposefully left unlabeled.

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

3.3.3 Optimization Procedure

All deep learning models in this section, both the SincIEEG described above and CNN model described in Section 3.3.4, use stochastic gradient descent from gradients produced by error back-propagation. We use the Adam optimizer [191] and fix the learning rate to $\alpha = 0.001$ for all experiments. Binary cross-entropy loss between the target label and the model's output is used as the objective criteria.

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

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

3.3.4 Experiments

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

Prediction Accuracy

The prediction accuracy is simply computed as the proportion of windows correctly classified as 'speaking' or 'not-speaking'. Visualizations that overlay the stimulus cue, curated labels, speech audio signal, and the model's predicted likelihood of speech are presented. Aligning recorded speech with model predictions across multiple training windows enables an examination of the model's predictions with both the labeled regions and recorded speech data. The model's ability to predict speech occurring outside the labeled region help to validate the model's generalization capabilities. Ultimately, this visualization provides an indication as to how the model would perform in practice. For instance, frequent oscillations in the predicted likelihood may achieve reasonable accuracy but ultimately be unreliable for use in a classification pipeline.

Spectral Band Convergence

A key aspect of this model's utility is its ability to learn spectral bands that minimize the loss function of the network. When the band parameters are combined with the loss and cross validation loss for each training batch, a visualization of the band convergence over time can be obtained. This visualization can serve several purposes. For the present analysis it serves as an additional method of model validation and interpretation. The model is explainable by design, allowing us to determine the frequency bands the model identified as empirically predictive. We can compare the frequencies used by the model to those highlighted in prior work to help confirm the model has discovered task-relevant correlations, improving trust and generalizability. For other analyses, it could serve as an exploratory tool to investigate whether frequency information is central to the phenomenon.

Comparison Models and Benchmarks

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

Auditory Cortex Electrode Removal To verify that classification performance was not merely being driven by auditory feedback, electrodes in the auditory cortex region were manually identified based on anatomical landmarks and removed from the analysis (see Figure 34). An abbreviated evaluation of SincIEEG was performed to confirm that the classification performance was not significantly degraded by the exclusion of the auditory electrodes. Optimization time of these additional models was reduced by using early stopping as described in Section 3.3.3. Additional testing verified that early stopping does not unfavorably bias the resulting model performance.

Linear Discriminant Analysis (LDA) and SVM Benchmarks: To explore whether the frequency bands that the SincIEEG model identified would confer some benefit over using the entire broadband spectrum, the performance using the bands that 3-band SincIEEG learned for each participant was compared to the performance using broadband activity from 0.5-170 Hz frequencies. The 3-band version was chosen to compare because it is more distinct from broadband than the 5-band version which generally occupies a greater proportion of the spectrum. A LDA and a linear SVM were implemented as performance benchmarks. Because these comparatively simple classifiers are not capable of attaining reasonable performance using raw ECoG timesamples, a preprocessing method derived from [192] was implemented that generates a band power aggregate measure over a 500 ms window that updates every 50 ms. The labels were accordingly downsampled to 20 Hz. For each label, the preceding 500 ms of the corresponding preprocessed ECoG signals were used to compute the input features. The resulting feature array was flattened into a vector for training the LDA and SVM models. This process was performed for both the broadband and 3-band SincIEEG versions.

Standard CNN: To establish how SincIEEG performs compared to a traditional deep learning method, a standard CNN was implemented and evaluated based on [193]. For this CNN, the first convolutional layers aggregate across time with kernels and stride of five samples, and a dilation of two samples to further downsample. The next layer maintains the kernel's size and stride, but returns to default dilation of one. The remaining two convolutional layers learn 3×3 kernels with unit stride and dilation until a final dense layer outputs to a sigmoid activation. A total of 16 filters were learned in each convolutional layer. The standard convolutional network model is an important alternative to SincIEEG as it uses the same convolution operation but is not directly interpretable. The training and testing paradigms remained unchanged, only the model architecture was exchanged.

3.3.5 Results

Prediction Accuracy

The average SincIEEG model accuracy across all participants was 94.1% (s.e 3.5%), and all but one participant achieved an accuracy above 90%. Figure 22 shows the accuracy of each hyperparameter configurations per participant with each configuration repeated three times. Results from Participant 1 and 2 were very consistent regardless of hyperparameter, while Participant 3 showed significant variability in the 3- and 5- band versions, and Participant 5 performed better without dropout. These differences are most likely mediated by electrode number and placement.

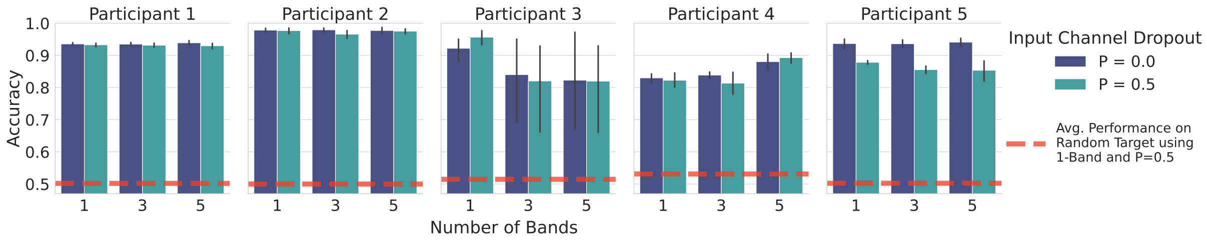


Fig. 22. Mean and variance of accuracy for all repetitions' test folds, for each participant model configuration.

However, the ability of the model to achieve good performance on such a variety of electrode locations is a testament to its robustness, and the advantages of a participant-specific feature set.

As described in Section 3.3.2, target labels were created from the timings of experiment cues, rather than the participant's speech. Therefore, to better gauge speech detection performance for practical speech detection applications, predictions were qualitatively assessed by visual inspection into one of three categories: *Full Success*, *Partial Success*, and *Failure*. A word trial was considered a *Full Success* if the prediction captured the entirety of the spoken word prior to onset and maintained until speech had ceased. Subplots (a), (d), and (g) in Figure 23 are examples of *Full Success* trials. Regions of false positive predictions encompassing a correctly identified speaking region were still categorized as a *Full Success* since false positives are envisioned to be less critical than false negatives for future applications to imagined speech. A trial was considered a *Partial Success* if it captured the majority of the word but clipped either the beginning or end. Subplots (b), (e), and (h) in Figure 23 are examples of *Partial Success* trials. A trial was considered a *Failure* if the word was missed entirely, if the model prediction was erratic or inconsistent, or if a portion of the word was missed from an otherwise well-placed detection. Subplots (c), (f), and (i) in Figure 23 are examples of *Failure* trials.

For each participant's best model configuration, we selected the model with the best cross-validation performance and assess its test-set predictions using the criteria described above. Table 8 shows the proportion of words assigned to each category for a 115 word test set for each participant for the respective best model configuration. Participant 1 and 2 models were able to very consistently predict speech before speech onset, suggesting that the model and electrode location combination may capture aspects of speech planning. Participant 3 and 4 models had a majority of partial successes. These trials largely exhibited clipping the beginning portion of words, suggesting that the model may be capturing aspects of speech production rather than speech planning.

Spectral Band Convergence

Figure 24 shows a representative example of spectral bands converging over training epochs. While there was a significant amount of variability in the plots across participants and configurations, there are several consistent observations. First, there is a distinct and consistent difference in the band evolutions during training when dropout is included in the model. With dropout, bands tended to converge more smoothly, rather than exhibiting large jumps in value as observed without dropout. With shared parameters, zeroing a sensor channel eliminates its influence and subsequently allows other sensors of varying magnitudes to drive parameter updates. Further-

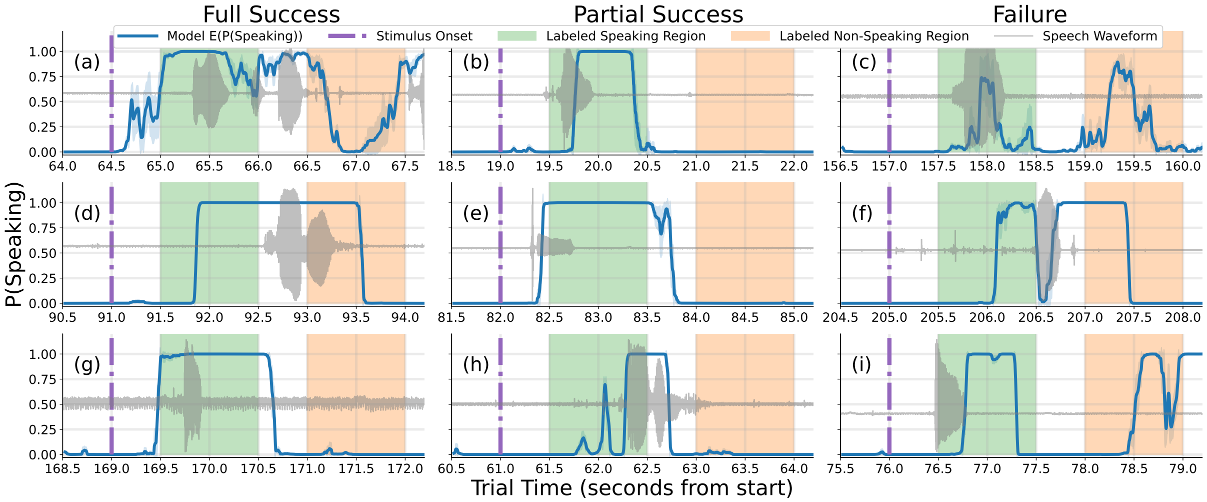


Fig. 23. SincIEEG model predictions of 9 representative words, grouped into 3 categories detailed in Section 3.3.5. The grey trace is the audio waveform from the microphone and represents the participants utterances during the word trial. The blue trace, and associated shading, represent the moving average and standard deviation of the model-derived 'speaking' likelihood over the previous 15 samples. The green shaded area represents the region labeled 'speaking', and the orange shaded area represents the region labeled 'not-speaking'. Top row: Participants 5, 4, 5. Middle row: Participants 1, 3, 2. Bottom row: Participants 3, 1, 2

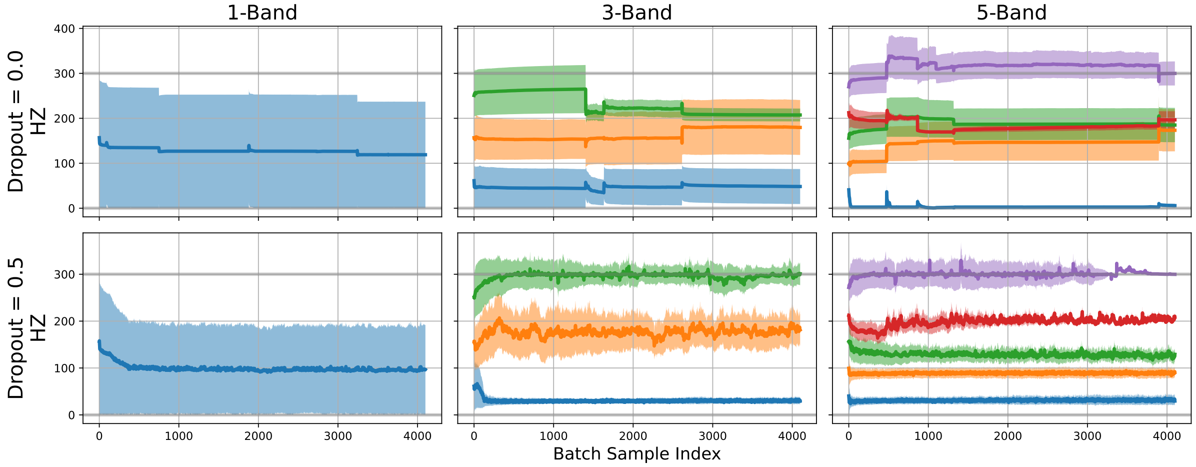


Fig. 24. Spectral band convergence of the 1-, 3-, and 5-band SincIEEG networks for Participant 2. The bold lines are the center of the band, and the shaded regions in the corresponding color are the band bounds. The top row is without dropout, and the bottom row is with dropout.

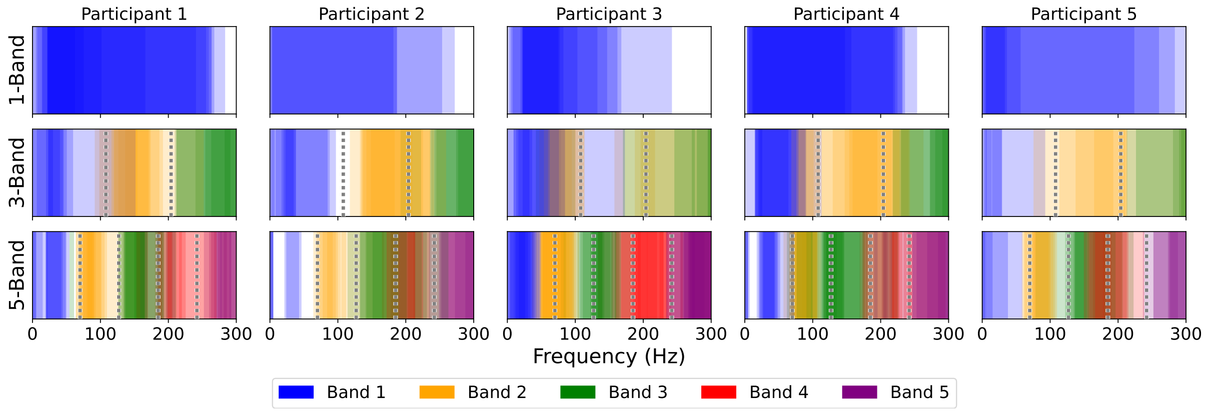


Fig. 25. Learned frequency bands for each the participant and each band combination. The selected bands are superimposed on a single frequency spectrum as a density plot at high transparency. Each band is plotted in a different hue: blue, yellow, green, red, and purple. More saturated hues represent frequencies common across more participants and configurations than less saturated frequencies. Vertical dashed lines correspond to the initial cut-off frequencies of adjacent bands prior to convergence. More details on the band initialization procedure can be found in Section 3.3.1.

more, zeroed sensors bias downstream normalization layer statistics towards zero. It is posited that these aspects result in the higher variance stochastic search of frequencies illustrated in Figure 24.

The final bands learned for each participant, aggregated across sessions and hyperparameter configurations, are shown in Figure 25, with the bands aggregated across participants shown in Figure 26. For better visualization, only SincIEEG models with performance in the top 50% for each participant are included in the figures. The bands are superimposed on a single frequency spectrum as a density plot at high transparency. Each band is plotted in a different hue, with more saturated hues representing frequencies common across more participants and model configurations than less saturated frequencies. This provides a compact conceptualization of the final converged frequencies across models.

For the 1-band case, the general tendency is for the band to be broad. However, the aggregated data shows that the bands commonly overlapped around 25-75 Hz, implying the lower frequency band may be more predictive than high gamma for the task as supported by [181].

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

A benefit of the interpretability of learning frequency band is that the results can be directly compared to known physiologically-relevant bands. Kanas et. al. examined 8 Hz wide frequency bands from 0 to 248 Hz, and produced a histogram ranking bins by contribution to speech detection [194]. It is a multi-modal distribution, with two larger peaks, one spanning 0-40 Hz and one 180-200 Hz, with two smaller, broader peaks in the intermediate frequencies.

Table 8. Prediction Success Over Trials

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

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

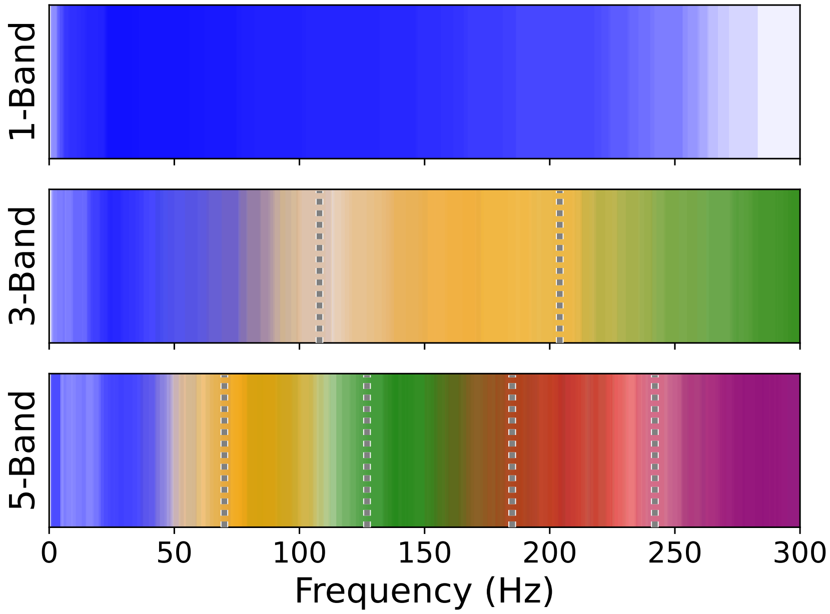


Fig. 26. Learned frequency bands for the top-50% of model configurations across participants for each band combination, as described in Figure 25. For improved visualization, the analysis only includes the top-50% of model configurations of each the participants' sessions.

Table 9. Model Accuracy Comparison

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

Comparison and Benchmarks

Table 9 shows the performance of all validation measures in comparison to SincIEEG. The SincIEEG and SincIEEG Non-Auditory results are the mean test fold accuracy for each participants' best performing hyperparameter configuration, effectively the highest bar for each participant in Figure 22. Excluding the auditory cortex electrodes did not significantly impact model performance. The causal formulation of the model, and accurate capture of speech onset within the predicted speech window, provides a strong indication that perception of speech was not a driver of the model classification accuracy.

The CNN architecture performance is overall on par with SincIEEG. This shows that the interpretable and parsimonious architecture of the SincNet does not compromise model performance.

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

It should be noted that, regardless of whether using learned bands or broadband, the LDA and SVM classifiers with the preprocessed ECoG signals did not achieve better results than SincIEEG. Additionally, SincIEEG was able to achieve better results with 30 times greater time-domain resolution than the methods using the preprocessed features.

3.3.6 Related work

In the last decade, neural speech decoding systems have made significant progress, including describing brain regions and mechanisms involved in speech, predicting words or phonemes, translating neural signals to articulatory kinematics models, text, or directly to speech waveforms [195, 189, 196, 197, 198, 199, 200]. Recent efforts have progressed to real-time decoding and synthesis of overt and imagined speech [192, 201, 202, 203, 204, 205]. While these studies primarily focus on broadband gamma activity (∼70-250 Hz), recent studies have shown that traditional lower-band frequencies (∼0-50 Hz) also contain relevant and complementary information for speech decoding [181].

Deep learning has been demonstrated to be an effective method for decoding speech from ECoG signals and its inclusion in the decoding and synthesis pipeline has increased in recent years [206, 200, 207, 203]. Although an end-to-end architecture may eventually be wholly effective with sufficient training data, some current approaches have adopted a modular scheme with several sequential component models, each configured for a specific aspect of the speech decoding process [208, 202, 203].

3.3.7 Discussion

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

In prior work on using ECoG for speech activity detection, Kanas et. al achieved maximum accuracies of 92% [208], and 98.8% with non deep learning classifiers[194]. Other studies used the detection model as part of a larger speech decoding analysis and so did not report specific results on speech detection performance [202, 203]. In comparison to SincIEEG, which uses unprocessed ECoG recordings, these approaches require appreciable signal preprocessing prior to speech detection. Since the feature extraction is inherent in SincIEEG, any latency introduced via explicit, potentially suboptimal, data-independent preprocessing is mitigated in the processing pipeline - which is critical for real-time implementation.

While we have demonstrated that SincIEEG is capable of speech activity detection from ECoG signals, the original implementation was used for acoustic speech detection [171], and it has also been applied to EMG signals [172]. Using a related approach for seizure detection using non-invasive EEG, Fukumori et. al. showed that a data-driven approach was superior to static filter banks [209]. Such models that learn the task-relevant spectral bands can be applied to other domains where frequency analysis is central. This is mainly due to the utility of learning bandpass filters, and the flexibility of the scope on which different filters can be learned.

In terms of interpretability, visualization of the learned bands provides a unique modality for studying the relevant spectral features. One consistent observation is that, across all band-number models and all participants, a low frequency component was always included in the models. This supports prior work that suggests lower frequency features can play a key role in speech detection in addition to broadband gamma [189, 194]. While the present analysis did not attempt to specifically identify the subset of electrodes related to speech production processes, due to the consistent performance results regardless of the hemisphere of the implant, it is expected that the contributions are largely from ventral primary motor cortex as shown in prior work [192, 199, 195, 210].

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

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

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

CHAPTER 4

ADAPTING FROM COHORTS TO INDIVIDUALS WITH TRANSFER LEARNING

4.1 Introduction

Classification of human activities from bioelectric sensor data is challenged by inter-subject variance and resource-constrained platforms, as discussed in Section 2.2. To address these issues in experiments, this work focuses on surface EMG data: electric potential of skeletal muscles sampled over time. Models in this domain must adapt to shifting sensors and new users while remaining simple enough to function on mobile devices. We contribute an approach to these challenges with *SincEMG*, a deep neural network that exploits digital signal processing concepts and transfer learning to reduce model size for activity recognition on raw sensor data.

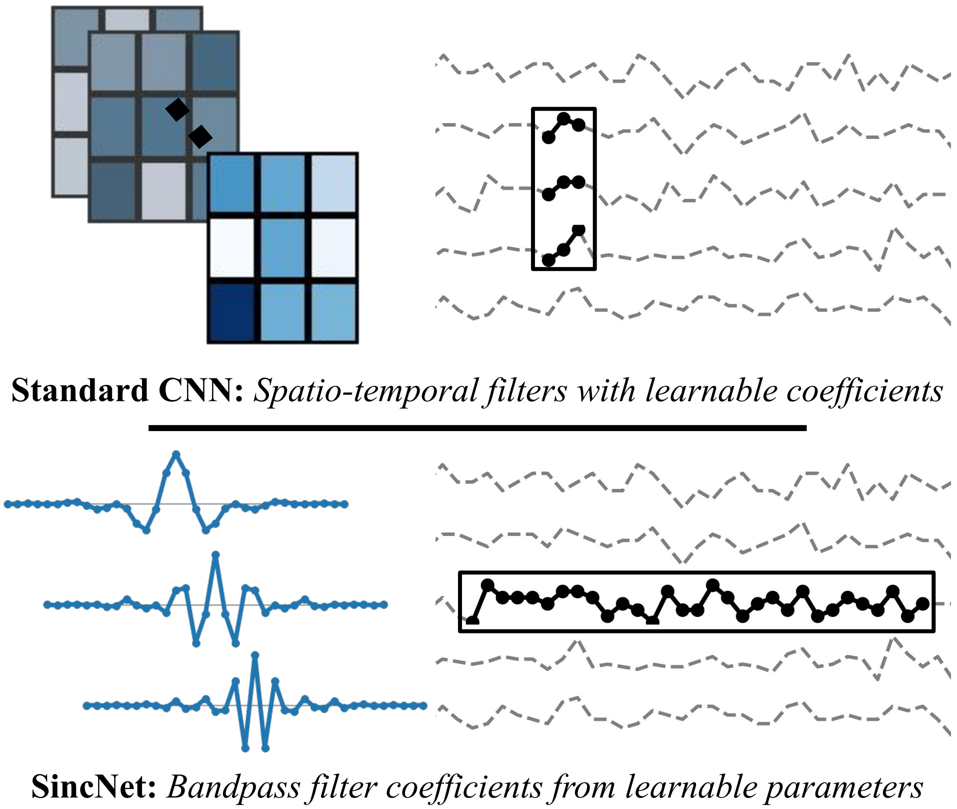
The model's first layer decomposes signals into frequency bands using FIR filters optimized directly from the data. The subsequent convolutional layers downsample across time and aggregate the first layer's band data. Batch normalization and dropout help to regularize intermediate layer outputs. This approach reduces compute requirements by decreasing the number of learned parameters and eliminating any significant data pre-processing. In addition to these improvements, the model's first layer learns a set of bandpass filters, which provide insight into predictive regions of the source spectrum.

4.2 SincEMG Model Architecture

Motivated by the need for smaller models with meaningful features, we propose *SincEMG*, a deep learning architecture for resource constrained activity recognition on EMG sensor data. The SincEMG model architecture is illustrated in Figure 28 and is composed primarily of convolutional layers, with a dense classification output layer.

For the initial layer, we implement Multi-SincNet as a multi-sensor extension to SincNet [212] to extract time-series features across an arbitrary number of channels. Applying DSP methods like those used in SincNet effectively establishes a prioi knowledge of sample inter-dependence within the input, improving predictive performance and reducing required parameters. The resulting bandpassed outputs are downsampled first using two layers of strided convolution through time. The 1-dimensional convolution reduces parameters while a stride reduces the size of the layer's output feature-space. Overall, these design decisions reduce parameters while helping to retain information that might be more easily lost in a pooling layer. Next, a cross-band convolution layer aggregates the channel's band data back to one signal before being passed to fully connected output layers. Batch normalization [182] is used between each layer after the initial bandpass layer.

The combined band extraction, strided convolution, and batch normalization operates on raw data to normalize features and produce a low-dimensional representation for classification. These model features are discussed in detail in the following subsections.



Standard CNN: *Spatio-temporal filters with learnable coefficients*

SincNet: *Bandpass filter coefficients from learnable parameters*

Fig. 27. A standard Convolutional Neural Network (CNN) learns coefficients of arbitrarily shaped filters (3x3 shown here) in each layer. The SincNet layer uses the *Sinc* function to derive one-dimensional bandpass filters from two learnable parameters. The above example illustrates three 31 coefficient band pass kernels, requiring a total of 6 parameters. This effectively embeds the model with a priori knowledge that the signals are time-varying with periodic components, greatly reducing model complexity while improving interpretability.

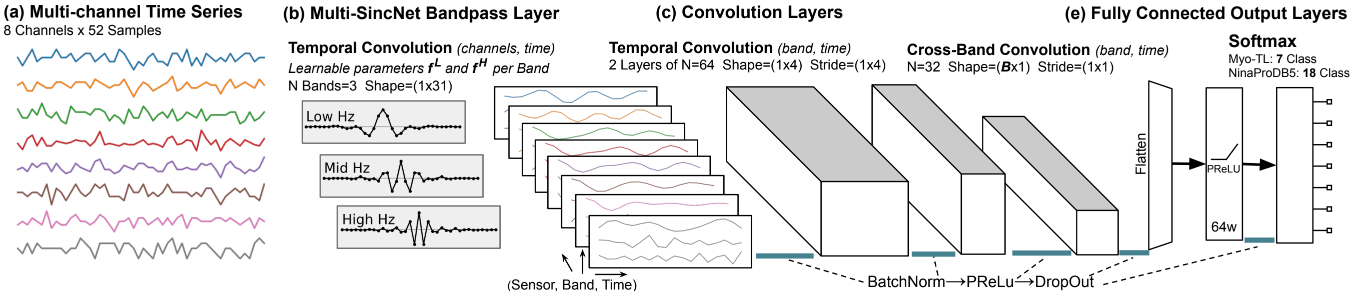

(a) Multi-channel Time Series
8 Channels x 52 Samples
(b) Multi-SincNet Bandpass Layer
Temporal Convolution (channels, time)
Learnable parameters f^L and f^H per Band
N Bands=3 Shape=(1x31)
Low Hz
Mid Hz
High Hz
(Sensor, Band, Time)
(c) Convolution Layers
Temporal Convolution (band, time)
2 Layers of N=64 Shape=(1x4) Stride=(1x4)
Cross-Band Convolution (band, time)
N=32 Shape=(Bx1) Stride=(1x1)
Flatten
(e) Fully Connected Output Layers
Softmax
Myo-TL: 7 Class
NinaProDB5: 18 Class
PReLU
64w
BatchNorm→PReLu→DropOut

Fig. 28. Example of our proposed architecture with three bands extracted across 8 channels (data shown taken from [211]). Raw data is passed into the Multi-SincNet layer, which extracts B bands per channel (three bands per channel shown in this figure). The band data is then downsampled by two layers of strided convolutions across time, followed by a convolution across bands. Values are finally passed to a dense layer block with a softmax output.

4.2.1 Band Extraction with Multi-SincNet

In order to abate noise and extract component signals, digital bandpass filters, such as FIR filters, can be designed to target informative areas of the signal spectrum. As shown below in formula 4.1, a discrete FIR filter kernel with coefficients k are applied to a time-series X using a one dimensional convolution.

$$X \otimes k_{(f_L,f_H)} = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} X[i] * k_{(f_L,f_H)}[j-i] \tag{4.1}$$

The number of samples $|X| = N$ must be large enough to support all source frequencies, but each new sample in a window increases output delay. In a similar way, more kernel coefficients $|k| = M$ can better approximate the idealized filter, but larger filters require more padding to fit on smaller windows of data, attenuating the output signal.

The values of k are determined analytically as described in [212] to behave as a bandpass filter $k_{(f_L,f_H)}$, rejecting all frequency components in X outside the region defined by cutoff frequencies f_L and f_H. We represent the derivation provided in [212] and implemented in [180] as a collection of kernels K, each computed from a differentiable function $G(f_L, f_H)$ during the forward-pass of training. The SincNet layer described by formula 4.5 has a configurable number of filter kernels $|K| = B$ computed from $G(f_L, f_H)$ applied across collection of learned parameters F_L and F_H.

$$F_L = \{f_0^L, f_1^L, ..., f_{i=B-1}^L\} \in \mathbb{R}^+ \tag{4.2}$$

$$F_H = \{f_0^H, f_1^H, ..., f_{i=B-1}^H\} \in \mathbb{R}^+ \tag{4.3}$$

$$\text{SincNet}(F_L, F_H) = \{k_{(F_L(i),F_H(i))}\} \ i \in B \tag{4.4}$$

$$\text{Multi-SincNet} = \text{SincNet}_c(F_L^i, F_H^i) \ i \in C \tag{4.5}$$

We extend the implementation provided in [180] to a multi-headed version we refer to as Multi-SincNet. We use Multi-SincNet in the first layer to learn either a channel-wise or shared signal decomposition. The Multi-SincNet layer applies a SincNet to each of the 8 separate data channels - if weights are shared, a single SincNet layer is applied iteratively on each channel and errors are accumulated. Using per-channel SincNet filters creates a separate SincNet for each channel position ($C = [1, 7]$). Per-channel filters requires a small number of additional parameters for the cutoff frequencies and also requires more kernel coefficient derivations after each weight update.

The SincNet parameters in [212] are initialized as mel-scale filter-bank to align with the fundamentals of human speech waveforms. For this work, we instead linearly distribute the bandpass filters across the 100Hz spectrum.

4.2.2 Single Dimension Convolution

After being passed through the Multi-SincNet layer, the output $X \in R^{B \times C \times N}$ is made up of C channels decomposed into B bands with N samples. To minimize parameters, we use a strided convolution along a single dimension in each layer. First, the time dimension is reduced by a 5 element kernel with a stride of 4 over only the time dimension. The time dimension is convolved twice since its the largest dimension, where $N > C$ and $N > B$ will be true in our experiments. Next, the channel bands are aggregated with a kernel sized to match the number of bands B and input padding. This spectral convolution layer collapses the band-level dimension. Finally, the output is passed to dense layer with 64 units followed by an output layer for class prediction.

4.2.3 Regularization

In order to avoid over-fitting we employ two common techniques - layer dropout [213] and batch normalization [182]. Dropout with a rate $p = .25$ is used after activation to discourage over-reliance on specific features within the layers. To address how magnitudes may shift over subjects, spatial batch normalization is used to scale all intermediate outputs after the Multi-SincNet layer.

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} \tag{4.6}$$

The model does not learn affine transformation parameters γ and β for batch normalization, but statistics $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ are collected during training for use during evaluation. These are kept with a momentum of 0.1 during both pre-training and subject fine-tuning.

4.2.4 Multi-SincNet Parameter Sharing

When applying multiple SincNet layers across channel inputs, the Multi-SincNet layer can be configured to use a unique set of parameters for each sensor, or a single set of *shared* parameters that are applied to each channel. When sharing parameters, the same set of bands are extracted from each channel, and error is accumulated from each channel onto the parameters through a summation. Sharing parameters therefore encourages the model to find frequency bands that extract informative features across all channels. If instead bands are optimized per-channel, then the model may more easily become over-fit to a particular sensor configuration or subject, reducing predictive performance.

When channel rolling (Section 4.2.5) is used with shared filters in the initial band pass layer, the regularization primarily impacts the ordering of features at the dense layers. In this case, channel rolling data augmentation is ensuring the global view does not become dependent on spatial location, while sharing band pass filters forces the model to ignore this augmentation at the model's input layer.

4.2.5 Channel Rolling

To prevent the model from learning a dependence on sensor orientation, we experiment with *channel rolling*, or *channel displacement*, during training. When channel rolling is enabled, input data is augmented by shifting values across the channel dimension by a random amount $r \in \mathbb{U}^{\mathbb{Z}}[0, |C| - 1]$, swapping each values' channel in a sample from X_c to $X_{(c+r) \bmod |C|}$ before being compiled into a batch. This type of data augmentation is conceptually similar to image rotation and translation used to regularize models for image processing: both data augmentation techniques model expected data transformations that are likely to be seen in practice, regularizing the model under optimization

The configuration of the input Multi-SincNet layer (Section 4.2.4) dictates how channel rolling samples during training effects the model. If the Multi-SincNet input layer is configured to learn per-channel filters, then the value of r alters the underling signal that the learned parameters operate. This change propagates through the model, forcing the entire model to adjust to possibly shifting distributions.

When the Multi-SincNet input layer is sharing parameters across each channel, channel rolling doesn't alter how the input layer processes data. In this case, the single set of filters are shared across each sensor - their location in the channel dimension does not map to a unique set of filters. The convolutional layers that follow are also unaffected by channel rolling in this case, since they operate along the time and band dimensions across each channel. However, the global view established at the dense layer is partly determined by the ordering of the sensor channels. Therefore, when the input layer is sharing parameters and data is augmented with channel rolling, the intermediate distributions will only shift at the input of the last dense layer. The resulting methodology prevents over-fitting by encouraging the model to learn a global feature extraction invariant to sensor location.

4.3 Experiments

Sensor data in HAR suffers from subject-level variance in features, which skews models that can't account for shifts in feature distributions between use. Shifting sensor locations and their conductivity similarly contribute to the challenges of HAR sensor data. Therefore, HAR models must be evaluated on datasets that include multiple users across several trials and tasks. This section first describes the public datasets used in our experiments, followed by details regarding experiments, model configurations, and data augmentation.

4.3.1 Transfer Learning

Differences in subject and training environment make optimizing a fully generalized model difficult. Instead, a common approach is to use a transfer learning scheme, wherein a model derives some generalization from a pre-training cohort before later being fine-tuned on a new

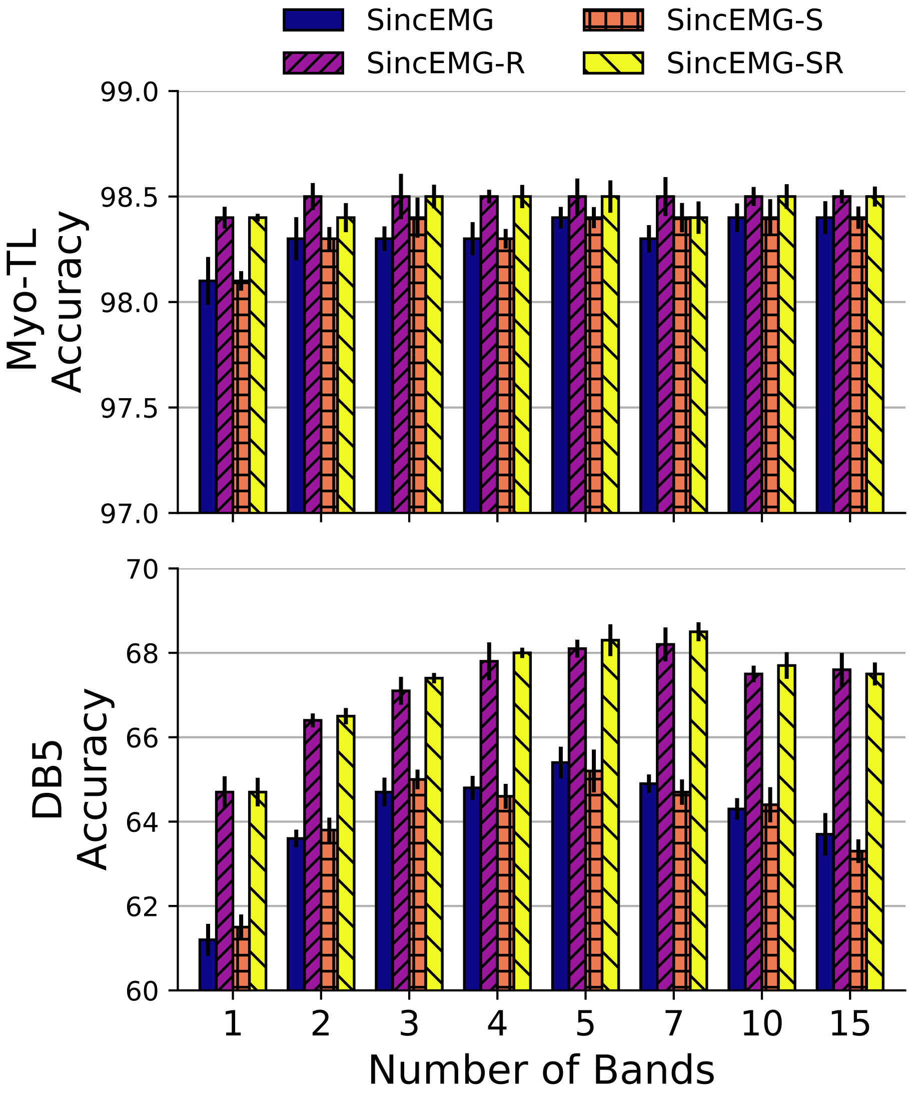


Fig. 29. Results on test data for each dataset across increasing number of bands B for different modeling approaches. ***SincEMG*** is our baseline model in which each channel has a unique set of learnable parameters in the initial Multi-SincNet layer. ***SincEMG-S*** instead uses a single set of *shared* learned parameters to bandpass all channels. The original and shared models are also evaluated with the channel rolling data augmentation, shown here as ***SincEMG-R*** and ***SincEMG-SR*** respectively. Note that DB5 models use a larger output layer with 256 units to help account for the increased number of output classes.

Table 10. Best performing model for each experiment model configuration (N=5). The overall best model and its results are bolded for each dataset.

Dataset	Model Output	Model Configuration	N Bands	N Params	Performance
Myo-TL	64 Units x 7 Classes	Per-channel Filters	15	58,779	98.409% (±0.110)
		Shared Filters	5	54,720	98.419% (±0.111)
		Per-channel Filters + Channel Rolling	7	58,918	98.523% (±0.027)
		Shared Filters + Channel Rolling	**5**	**38,069**	**98.524%** (±0.038)
DB5	256 Units x 18 Classes	Per-channel Filters	5	54,790	65.450% (±1.321)
		Shared Filters	5	57,236	65.182% (±1.218)
		Per-channel Filters + Channel Rolling	7	58,918	68.166% (±1.154)
		Shared Filters + Channel Rolling	**7**	**58,820**	**68.456%** (±1.209)

subject. To perform transfer learning, the dataset is first separated into a pre-training portion and an evaluation portion. A base model is then optimized using the pre-training portion. Once complete, the model is optimized, or "fine-tuned", on the train partition of the subject whose under evaluation. The evaluation subject represents a new user of the system, whom our model must adapt to during setup. Once fine-tuning is complete, the model is evaluated on the subject's holdout partition, simulating use in application.

Myo-TL provides the data partitioned into pre-training subjects and evaluation subjects. We use the pre-training subject data only for pre-training, then we fine-tune the model on each evaluation subject by copying the pre-trained model and training with the evaluation subject's data. The first test partition for the evaluation patient is used for cross-validation and performance monitoring. The second test partition of each evaluation subject is reserved for performance comparison.

DB5 is not designed for transfer learning experiments, but we use a procedure matching [211] by using a leave-one-subject-out K-Fold scheme. Thus, 9 training subjects in each of the 10 folds are used for pre-training, and the remaining subject is used for fine-tuning and evaluation. Each subject has 6 sessions, which we split into partitions of 4-1-1 for training, cross validation, and testing. Within each fold, the 4 training sessions of the 9 pre-training subjects are used together for optimization and their single cross validation sessions are combined to examine model performance for checkpoints. After pre-training, the model is fine-tuned on the held out evaluation subject's 4 training sessions. Similar to before, the cross validation session is used for checkpoints and the test session is used for performance comparison.

Using the DB5 and Myo-TL datasets, we examine our architecture for use in hand pose classification with a low-cost EMG wearable device. Models are built using the transfer learning scheme described in Section 4.3.1, the performance of each model is reported as the mean across all evaluation subjects. Final model performances reported are each the result of 5 repeated experiments.

Models are optimized using stochastic gradient descent for 100 epochs on the pre-training portion of the data. Cross-entropy loss is weighted by $w_i = \frac{1}{E(y_i)}$ where $E(y_i)$ is the rate of class i in the training data. Performance is monitored using the cross-validation data. The best model, according to cross validation loss, is stored for the evaluation stage. After pre-training has completed, the checkpoint of the best performing model is restored, and the model parameters are overwritten. The pre-trained model is then applied to the training data of the evaluation subject. The optimizer and other model statistics are reset before training in the fine-tuning phase proceeds for another 100 epochs. The model parameters with the best cross validation performance during fine-tuning are again saved for restoration at the end of the 100 fine-tuning epochs. For all models, the performance on test data by the model restored from the best fine-tuning state is the performance we report.

Models are implemented using the PyTorch deep learning and tensor framework [214] and are optimized through gradient descent using the Adam optimizer [191]. Weights in all layers except the Multi-SincNet layer are initialized from the normal distribution. The Multi-SincNet is initialized as described in Section 4.2.1.

4.3.2 Classification Performance

The best performing model for both datasets utilized shared parameters combined with channel rolling for improved generalization. The Myo-TL dataset required only 5 bands for its best performance, reaching 98.524% accuracy with only 38,069 parameters. We find the DB5 dataset to be more challenging since it has fewer subjects and far more classes than the Myo-TL dataset. Still our model exceeds the accuracy of previously reported results using 58,820 parameters to achieve 68.456% accuracy on the DB5 dataset.

The mean (N=5) confusion matrices and per-class F1-scores for SincEMG-SR-5 are shown in Figure 30. Myo-TL's smaller number of classes and increased data produces F1 scores above 0.95 for all classes. We find significant confusion between *hand open* and both *ulnar and radial deviations*. In contrast, DB5's reduced dataset size and increased number of classes results in F1 scores for SincEMG-SR-7 largely below 0.75 per class.

Several groupings of misclassification can be found in DB5's confusion matrix in Figure 30. The *abduction of all fingers* (i.e., spreading fingers apart in parallel with hand) tends to be misidentified with a *thumb opposing little* (i.e., pinching with thumb and little finger). Pronation (i.e., palm facing down) and supination (i.e., palm facing up) through rotation of the wrist are often conflated.

4.3.3 Feature Interpretation

An important benefit of using the Multi-SincNet layer is that the model learns interpretable parameters in the first layer. The parameters represent a set of band pass filters, highlighting portions of the spectrum in the input utilized by the model. Depending on the application, the parameters can be used to validate performance, and guide the design of other algorithms or research. Figure 31 and 32 plot the learned bandpass parameters over the training epochs for a model of the best proposed architectures.

In Figure 31, a SincEMG-SR-5 model is trained using the methodology describe in 4.3.1. Myo-TL pre-training reaches its best in under 20 epochs, delineated in the plots as the dashed vertical line. The larger versions of this model tested in this work achieve a better fit to pre-training data over more training epochs, but inevitably tend to overfit, or at least not improve, when fine-tuned for evaluation. Figure 32 shows four random subject folds from the training procedure describe in Section 4.3.1. The procedure results in a unique pre-training and fine-tuning phase for each of the 10 subjects in DB5. In both datasets, the bandpass parameters make coarse grained adjustments early in the pre-training phase, but the DB5 dataset continues to make clear adjustments in bandwdith and frequency late into the pre-training phase. Of course, fine-tuning in both datasets also shows larger changes to the parameters to more closely align with the evaluation subjects characteristics.

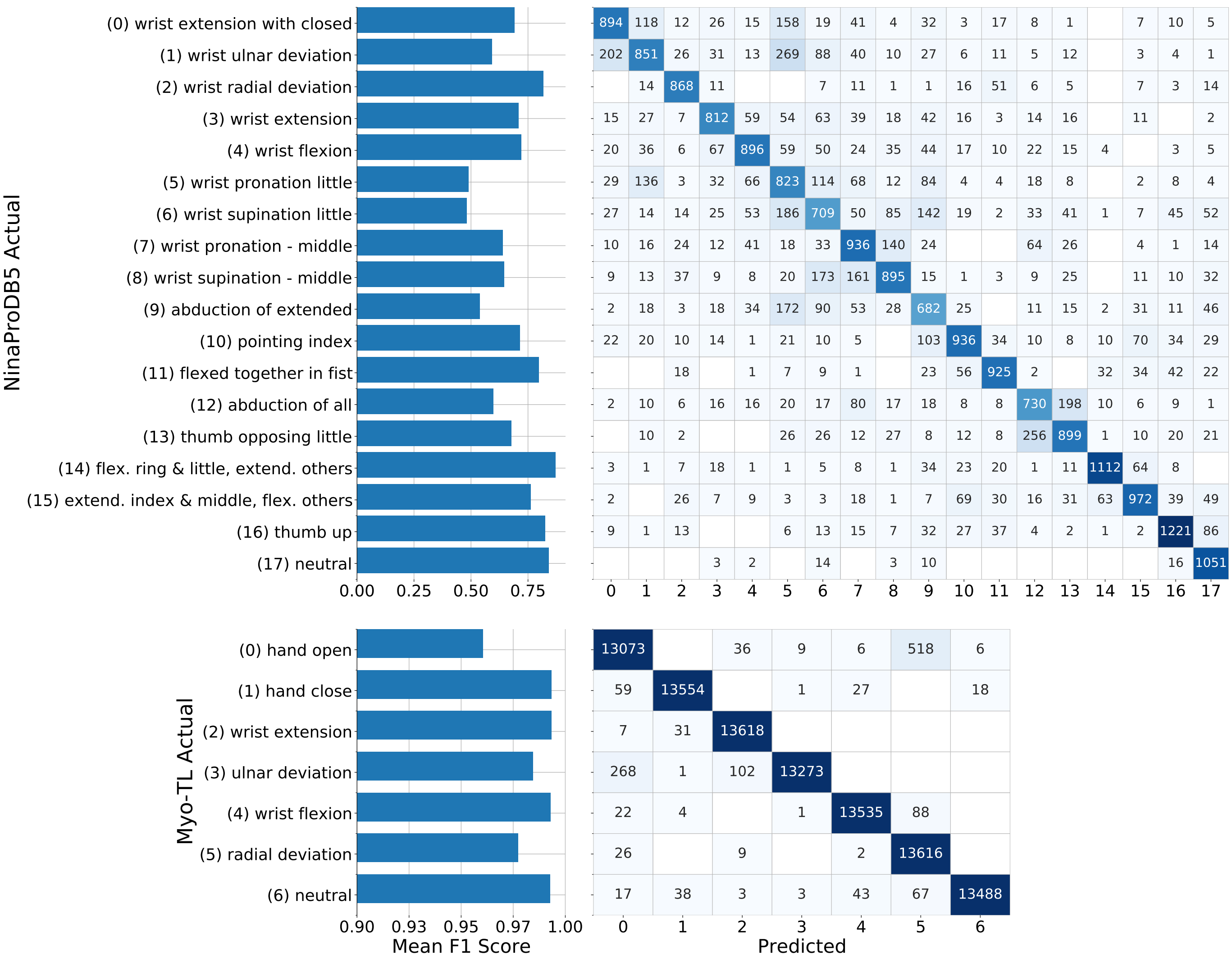


Fig. 30. Test partition confusion matrices for SincEMG on both the Myo-TL [215] and DB5 [216] datasets. Confusion matrices are the averages across each experiment (N=5) of the selected model from Table 10 rounded to the nearest whole integer. Mutli-class F1 scores are calculated by an unweighted average of each class-specific F1 score, then each test partition's class F1 scores are averaged across the repeated model experiments.

4.4 Related work

HAR enables rich user experiences for many applications, with the potential to improve the livelihood for persons with disabilities [87][88]. Increasingly, HAR approaches rely on mobile or IoT [83] devices equipped with sensors continuously monitoring the subject. In order to expand practical use-cases, recognition using this data is performed on-device, which minimizes response time and eliminates reliance on external systems [91]. However, mobile host systems are typically resource-constrained, often requiring low power and reduced weight in comparison to traditional host systems. Thus, activity recognition models are motivated to reduce model complexity while still maintaining acceptable recognition performance.

In pursuit of improved performance, popular deep learning [28] architectures are applied to sensor data for activity recognition [29]. Architectures such as CNNs [30][31][32], recurrent neural networks RNNs [33], or a combination [34][35][36] have been used to classify activities from sensors. These models achieve state-of-the-art results, but often require upwards of hundreds of thousands of learnable parameters. To reduce processing overhead, some efforts use general-purpose model reduction techniques [37] or avoid raw data entirely[38]. Furthermore, deep learning classification models borrowed from other domains, such as CNNs designed for image classification, do not leverage the unique characteristics of sensor data for HAR. Sensor data evenly sampled over time yields a time-series dataset, a data modality whose covariance and assumptions diverge from image data.

The authors of [217] survey recent deep learning efforts and the their application to large EMG datasets. Models in this domain may operate directly on raw samples, pre-process data into spectrograms or wavelet topographies, or utilize manually extracted features. The authors conclude with a discussion that notes the lack of research addressing the computational burden of deep learning methods applied to EMG data. To approximate computational requirements, this work will relate the number of learned parameters required by models. The number of learned parameters correlates with the computational demands since increased parameters require more

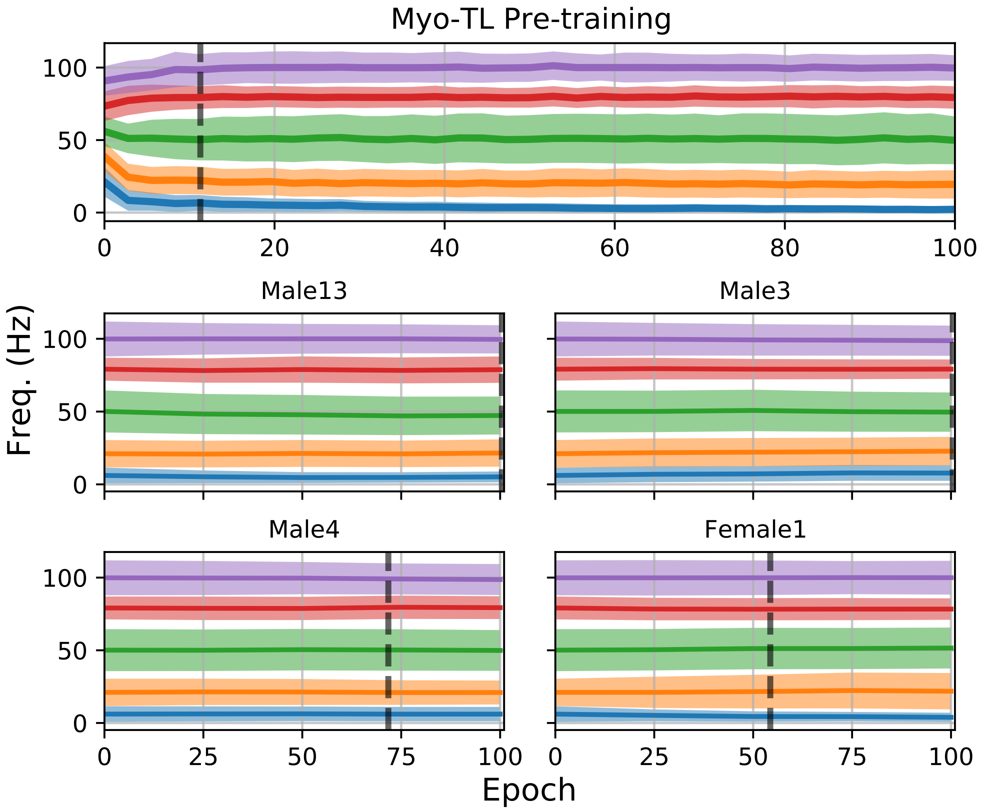


Fig. 31. Band pass regions for 5 shared filters over the per-subject training batches for the Myo-TL [215] dataset. The vertical grey line indicates the epoch where the snapshot of the best performing model was captured based on subject CV data.

storage and typically more operations to produce a prediction. However, this simplifying assumption overlooks derived or non-learned parameters.

Transfer learning using deep learning is explored in [211] for pose classification using EMG sensor data. The authors explore three primary models across two datasets, resulting in six model architectures. Their most effective model operated on wavelet features extracted before being passed to a CNN with dense outputs. Model variations in this work that process raw data directly require over 500,000 parameters. Their wavelet-based models require approximately 30,000 parameters.

Work in [218] used RNNs to classify windows of EMG data from the Myo band. Given the

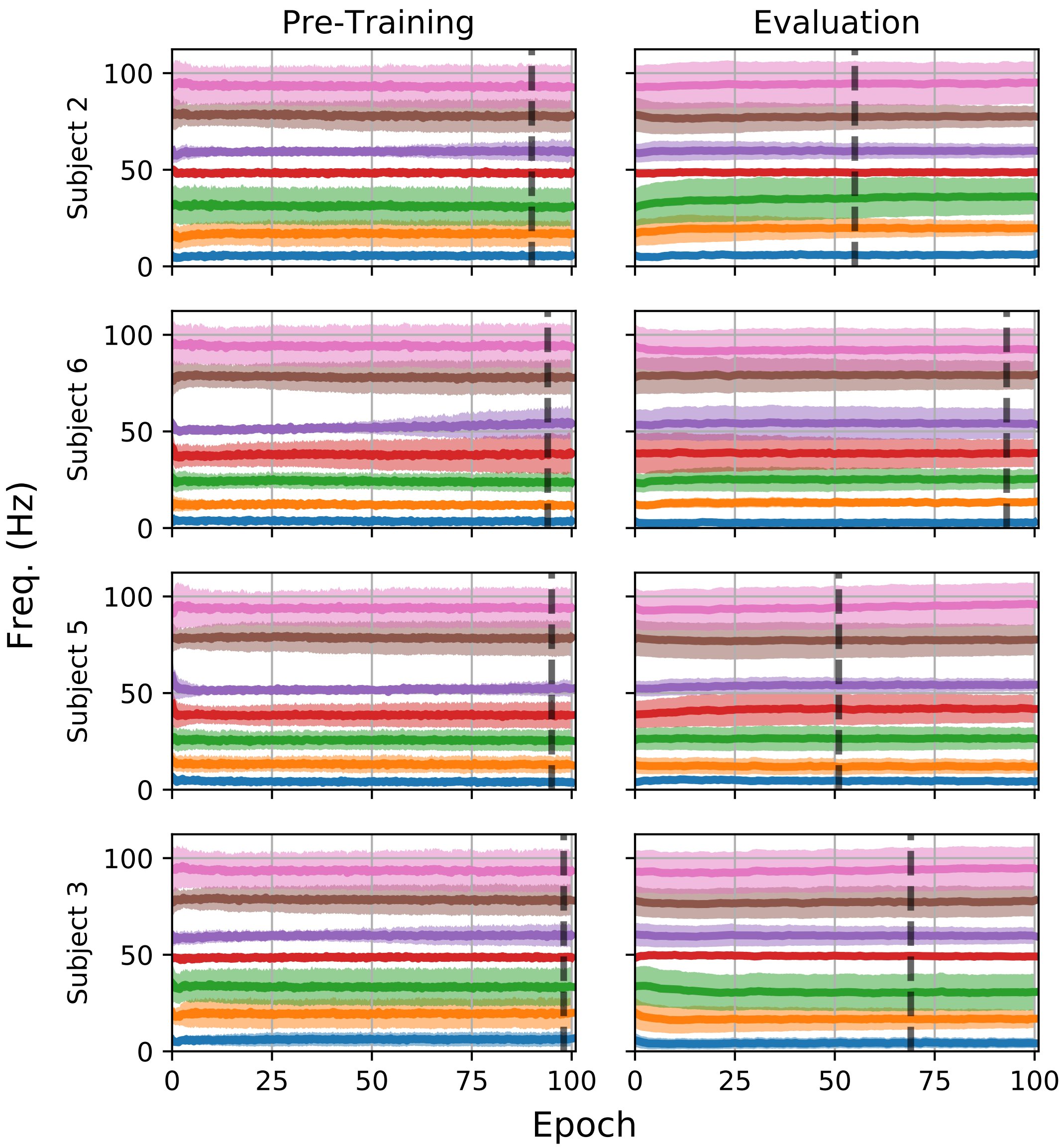


Fig. 32. Band pass regions for 7 shared filters over the per-subject training batches for the DB5 [216] dataset. The vertical grey line indicates the epoch where the snapshot of the best performing model was captured based on subject CV data.

use of gated recurrent units (GRU), we estimate their model uses at least 250,000 parameters. Experiments using grouped k-fold yields 77% classification accuracy. Many other efforts have explored CNN classification performance on EMG data. Intention is decoded from EMG data with sub-sampling and CNNs in [32], improving on support vector machine baselines. In [219], long-term EMG data is studied using data from daily experiments over 15 days, with stacked sparse autoencoders of handcrafted features and CNNs performing best. A regression a CNN is performed in [220] to predict hand orientation and pose. Sensor EMG spectrograms are used as CNN inputs for multi-task classification in [221]. SqueezeNet architecture is adapted in [37] to reduce the CNN size, resulting in only 5,889 parameters achieving an accuracy of 84.2% using a Myo band.

4.5 Discussion

Our results illustrate the utility of domain specific architectures and regularization. Comparing our results with prior work [211] in Table 11, SincEMG achieves competitive performance without pre-processing while using fewer parameters. Our best model for Myo-TL, SincEMG-SR-5, uses shared filters, channel rolling, and five bands to achieve 98.52% accuracy on raw input data. The most comparable model is CA-Raw, which requires over 500k parameters, compared to SincEMG-SR-5's 38,069 parameters. Otherwise, CA-CWT requires only 30,219 parameters, but uses inputs pre-processed into wavelets and ultimately performs worse than SincEMG-SR-5 on both datasets. SincEMG-SR-7+ achieves the best accuracy on DB5 using 58,820 learnable parameters. Increasing the number of bands to 7 and the dense layer width to 256 (denoted by '+') helps compensate for the increased number of classes.

Table 11. Comparison of the proposed SincEMG model with prior work [211]. For each dataset, the best model and its results are bolded.

Model	Pre-proc.	Best N Params	Myo-TL	DB5
CA-Raw	*None*	549,091	97.39%	68.08%
CA-Spectrogram	FFT	67,179	97.85%	65.10%
CA-CWT	CWT	30,219	98.31%	65.57%
SincEMG-SR-5	***None***	**38,069**	**98.52%**	-
SincEMG-SR-7+	***None***	**58,820**	-	**68.46%**

Scaling for increased classes is accomplished simply by adding additional band pass filters at the input and increasing the output layer width. In the case of the DB5 dataset, the increased capacity and reduced dataset size makes regularization an important component. Applying channel rolling regularization consistently improved accuracy. Our best models on DB5 further reduced parameters and improved performance by sharing band pass filters across sensor channels.

Channel rolling regularization is intended to prevent the classification layers at the output from relying on the position of a common signal, since it is unlikely to occur in exactly the same position across subjects. Unlike applying dropout to the channels, channel rolling does not remove information. When applied with shared band pass parameters, it only discourages over-fitting to the *spatial* location of the information. Combining channel rolling with per-channel filters in SincEMG similarly dissociates each unique filter from a specific spatial location. Effectively this creates a model that has increased feature extraction capability, but the data augmentation combats over-fitting.

The use of FIR band pass filters as the basis of a model simplifies implementation as in low-power application specific integrated circuits. However, in cases of extreme constraints, model selection may need to favor designs with a lower number of decomposing band pass filters. For example, remote sensors in areas with high-risk for hardware loss may wish to minimize hardware costs of sensors. Additional power and cost savings can be earned by reducing the amount of data transmitted. In this case, a SincNet-based model can be trained offline, using desktop and server class machines. The coefficients of the learned band pass filters are then included in the firmware for a low-power DSP micro-processor. For instance, the Texas Instruments TMS320C6x line of DSP chipsets[1] is well-suited for filtering using bandpass filters learned by the proposed SincEMG models. Such a processor can apply the filters on-device, reducing the input size before transmitting the intermediate results. Servers can then further process the data or store for later use.

This chapter presents a deep learning architecture for resource-constrained classification of time series sensor data. Inspired by successes in the speech domain, the SincEMG deep learning model operates on raw sensor data and yields smaller models with interpretable parameters. The results across two EMG pose classification datasets demonstrate how this architecture can achieve state-of-the-art results, even when data is scarce and the number of classes increases. Deep learning architectures have demonstrated impressive results in many domains, including visual and audio classification. Departure from the data center to resource-constrained domains, while challenging, is becoming more common for deep learning systems. As more domains leverage these models, we expect a continuous need to integrate prior knowledge for improved resource efficiency.

[1] https://www.ti.com/processors/digital-signal-processors/c6000-floating-point-dsp/products.html; Accessed October 24th, 2020.

CHAPTER 5

ADAPTING FROM UNLABELED COHORTS TO INDIVIDUALS WITH SELF-SUPERVISED LEARNING

5.1 Introduction

The contributions presented in this chapter approach many of the essential requirements related to adaptable and trustworthy HAR that we introduced in Sections 2.1 and 2.2. Specifically, model development of HAR approaches are challenged by costly data collection requirements - participants must perform the activities in real time, and the activity must be carefully aligned with labels for supervised training. Furthermore, bioelectric sensor data often has large discrepancies between individuals, typically requiring model developers train a model per each individual user, as we did in Chapters 3 and 4. Finally, it is often unclear the risk involved when contributing bioelectric sensor readings of an individual's physiology to a model's development.

In this Chapter, we first introduce a method for adaption to new users and tasks, and then further experiment with the method to quantify the risk of pretraining cohort privacy violations. Our methods are applied to problems related to speech neuroprostheses, which are systems designed to decode and synthesize speech directly from the electrical potentials of the brain. In Section 5.2 of this chapter, we introduce *brain2vec*, a Self-supervised Learning (SSL) methodology for representation learning of intracranial neural recordings. Our method does not require labeled data and supports combining data from any number of participants. The pretrained model can then be applied to a specific users tasks and data. Our experiments demonstrate brain2vec's ability to learn useful features for each participant's downstream tasks.

In Section 5.3 of this chapter, we extend our experiments to include pretraining on single participants and pairs of participants. We also propose adversarial threat models for potential information leakage from a brain2vec model. Our experiments suggest that models such as brain2vec can be used to train re-identification models, and that brain2vec's objective criteria can be used to perform membership inference against the pretraining cohort.

5.2 Self-Supervised Learning of Neural Speech Representations from Unlabeled Intracranial Signals

In this section, we present *brain2vec*, a sensor-level feature learning methodology that builds on recent progress by utilizing self-supervised pretraining, vector quantization, and spatio-temporal positional encoding for use in speech neuroprosthetics. We adapt semi-supervised Natural Language Processing (NLP) techniques to allow pooling of data across participants by re-referencing electrode locations of different participants to a common brain atlas before training. The proposed framework is used to pretrain a sensor-level feature extraction model on unlabeled data from multiple participants. For evaluation, the pretrained model is used to extract features for an unseen participant's speech related classification tasks. Importantly, the pretrained model's parameters are not updated to accommodate the new participant's data or sensor configuration, forcing the fine-tuning classifier to rely only on the features learned from pooled participant data. We also perform exploratory dimensionality reduction and visualization of the learned features to illustrate class separation for the downstream classification tasks.

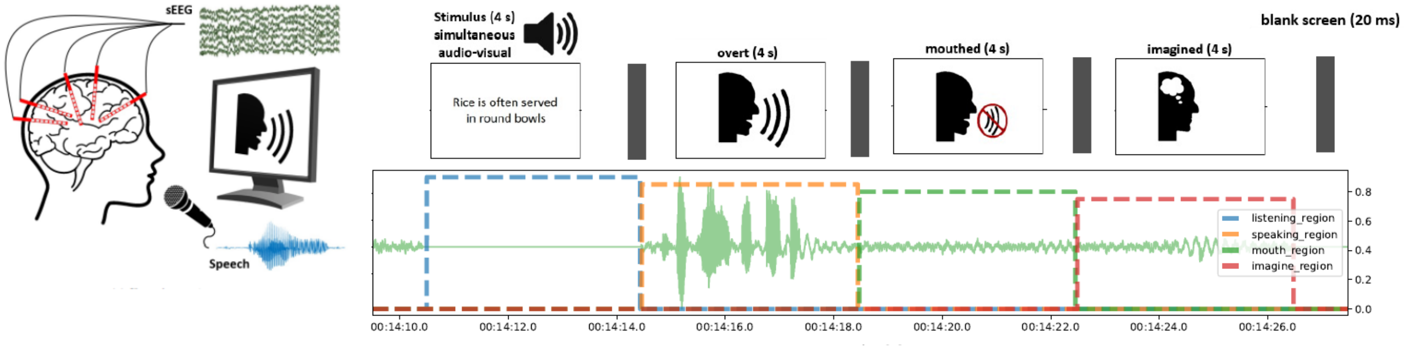


Fig. 33. Diagram of the Harvard Sentences experiment protocol. Detailed in Section 5.2.1.3.

Our results demonstrate that brain2vec is capable of encoding rich speech representations which can be used for classifying an array of disparate speech-related downstream tasks. These results show promise for a future in which "off-the-shelf" pretrained speech neuroprosthetics

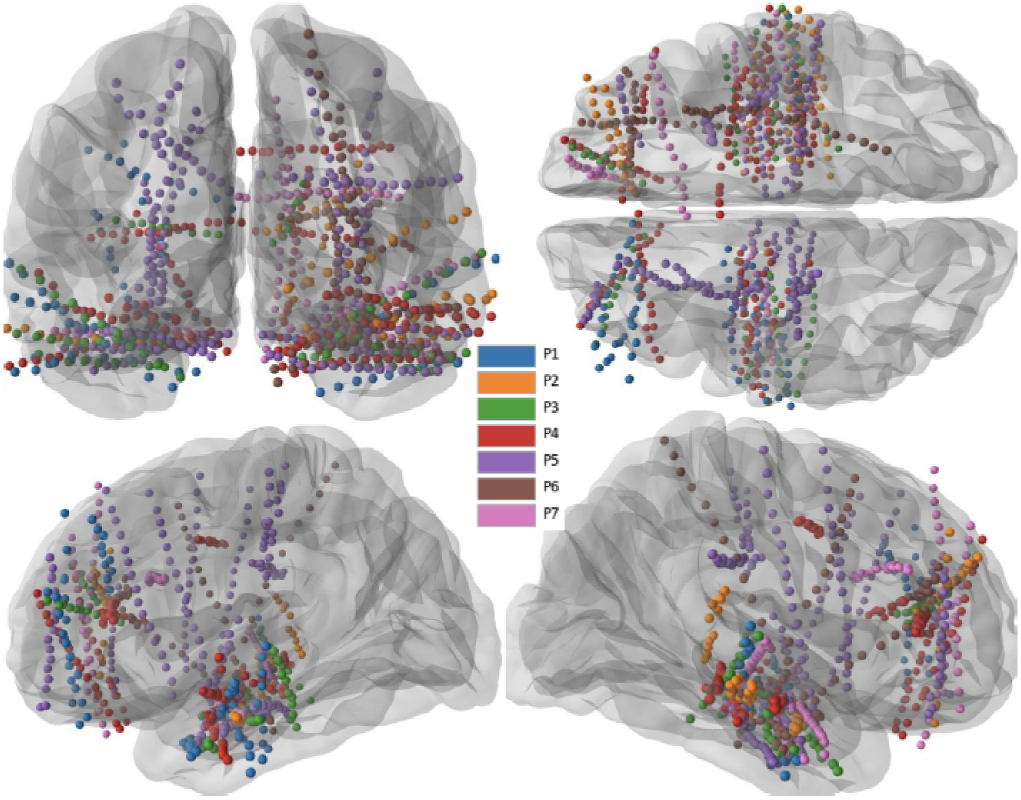


Fig. 34. Common atlas electrode locations for the 7 participants.

models can be used to improve a user's livelihood without the need for extensive data collection and labeling.

5.2.1 sEEG Data

To assess our method, we utilize data collected from seven participants, with time-aligned labels of speech behavior from an experimental protocol. This section describes our data and how it was collected.

5.2.1.1 Participants

sEEG data were collected from 7 native English-speaking participants being monitored as part of treatment for intractable epilepsy at University of California San Diego Health. The locations of sEEG electrodes were determined solely based on the participants' clinical needs. The number of implanted electrodes for each participant are provided in Table 5.2.1.1. The study was approved by Virginia Commonwealth University and UCSD Health IRB.

Participant	# Electrodes
1	90
2	70
3	80
4	175
5	232
6	94
7	108

Table 12. Number of implanted electrodes for each participant.

5.2.1.2 Acquisition Configuration

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

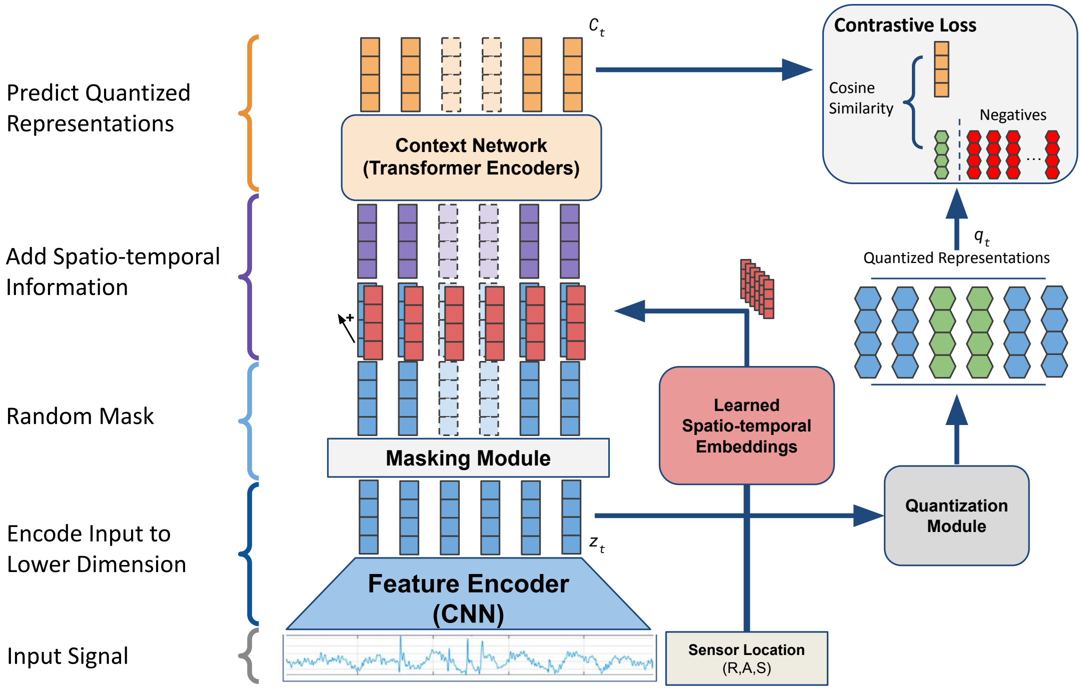

Predict Quantized Representations
Add Spatio-temporal Information
Random Mask
Encode Input to Lower Dimension
Input Signal
c_t
Context Network (Transformer Encoders)
Masking Module
z_t
Feature Encoder (CNN)
Learned Spatio-temporal Embeddings
Sensor Location (R,A,S)
Contrastive Loss
Cosine Similarity
Negatives
q_t
Quantized Representations
Quantization Module

Fig. 35. The brain2vec pretraining architecture that learns sensor-level representations. A 0.5 s window of normalized Stereotactic Electroencephalography (sEEG) for a single electrode signal is passed to a CNN feature encoder producing latent representations (blue). Spatio-temporal embeddings are created using the 3D Right Anterior Superior (RAS) coordinates of the electrode (red). The latent representations are from the feature are sent to the quantization module. The latent representations are then passed to the masking module, and then the positional embedding is added to the masked latent representations (purple). The embedded latent representations are the passed to the context network, which is a set of transformer blocks, which finally produce the context representations. The reconstructed context representations corresponding to the masked latent representations are then compared to the quantized vectors using cosine similarity in a contrastive loss paradigm. Further details of each component are in Section 5.2.2

5.2.1.3 Data Collection Protocol

The experimental protocol is designed to investigate overt and imagined speech processes in the brain by having participants repeat a sequence of sentences, each in a series of three different speaking modes. Before the experiment, the participant is explained the paradigm, experimental icons and cues, and instructed to perform the associated tasks immediately upon cue presentation - within a 4-second interval during which the task cue is displayed. A trial begins with a short sentence displayed on a computer monitor while simultaneously narrated through computer speakers. All sentence audio was less than 4 seconds in length, but regardless of the length, the associated text remains on the screen for 4 seconds. Following a 20 ms blank screen, the participant is cued with an icon to vocalize the sentence (i.e., overt mode), and this cue remains on the screen for 4 seconds. Following a 20 ms blank screen, the participant is cued for 4 seconds via icon to articulate the sentence as if they were speaking, but without vocalizing (i.e., mouthing mode). Finally, after a 20 ms blank screen, the participant is cued for 4 seconds by icon to imagine speaking the sentence without articulating or vocalizing (i.e., imagined mode). Then following a 20 ms blank screen, the next sentence trial begins. This protocol is illustrated in Figure 33.

The paradigm is repeated each time for a set of 50 unique Harvard sentences, designed to be phonetically-balanced conversational English [222]. All participants completed the entire set of 50 sentence trials; however, only 25 sentence trials from Participant 1 are evaluated due to a software issue that corrupted the labeling of the other 25 sentence trials.

5.2.1.4 Volumetric Morphing of Electrode Locations to a Common Brain Atlas

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

The 3D electrode coordinates reconstructed from Computerized Tomography (CT) and MRI imaging data can not be directly compared across participants due to anatomical brain differences. For this reason, each participants' electrode locations were converted from their native brain space coordinates to corresponding locations on the MNI305 common brain atlas [223, 224]. The mapping was done using the Freesurfer software package [225] and MNE-Python python package [226], where further information on the details of the affine transformation procedure can be found [225, 227].

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

Figure 34 shows the locations of all participant electrodes on the common brain atlas. Each electrode is represented using a 3-dimensional vector indicating its location on the common brain atlas. These coordinates are given in the RAS frame, with positive values in the 3 dimensions referring to right vs. left, anterior vs. posterior, and superior vs inferior, respectively. The coordinate units are in meters, and take on a range of values $[-0.076\text{ m}, 0.079\text{ m}]$ across all dimensions. The origin is located at the Anterior Commissure, and the negative y-axis passing through the Posterior Commissure.

5.2.2 Self-supervised Pretraining Methodology

Our primary contribution is a model architecture and pretraining methodology for learning generalized feature representations of brain activity, using only unlabeled sensor data pooled from an arbitrary number of participants. We refer to this approach as brain2vec, and this section describes the underlying model, loss functions, and optimization procedure. We later show in Section 5.2.3 that representations learned by brain2vec can be used to train classifiers on an array of labeled downstream tasks. Importantly, the brain2vec pretraining methodology enables fine-tuning on any number of sensors, including new configurations on unseen users.

The model consists of a sensor-level feature encoder, implemented as a CNN. The feature encoder's outputs are then passed to a transformer network that learns a latent context vector representation of the input sEEG signal. During the pretraining phase, the model is tasked with reconstructing masked regions of the input signal's latent representations, using self-supervised techniques pioneered by language models [228, 13, 14, 229]. The training is aided by a vector quantization module that discretizes the targets, thus guiding the network to learn a constrained hidden representation. RAS coordinates are used to learn a spatio-temporal embedding that is added to the input of the context model. The resulting sensor-level model can then be used for feature extraction in a task-specific fine-tuning procedure.

5.2.2.1 Model Architecture

The brain2vec architecture is based on the wav2vec2 audio modeling architecture [229], but with significant modifications to support the modality of intracranial sensor data, including changes to the feature encoder CNN, positional embedding paradigm, codebook configuration, and context network size. In this section, we first overview the input data and the key processing steps across the model's components. Further details on how brain2vec differs from wav2vec2 are described in each subsection.

Brain2vec's input is an unnormalized 0.5 second segment from a single sEEG channel. The input window is first downsampled to 512 Hz and standardized to a zero mean and unit variance within the half second window. The segment is then passed through a CNN-based feature encoder that generates the latent representations. These latent representations are then passed to both the Quantization Module, where they are discretized into a codebook vector for the objective function, as well as to the context network. The context network is a standard transformer architecture, producing context representations from the codebook distribution. Before entering the context network, regions of context representations across time are masked from the context network by replacing the context representation with a learned mask embedding. Then, spatio-temporal positional information is embedded in the latent representations before being pass to the context model. The masked context representations are learned by having to correctly choose their corresponding quantized latent representation from a set of distractors.

The decision to use a 0.5 s window was driven primarily by prior work, and the intuition that the majority of pertinent information for decoding speech from neural signals will be encapsulated in the neural activity immediately preceding the produced speech. In [230], a speech re-synthesis task was shown to be largely dependent on only 400 ms of neural data centered at the corresponding 400 ms audio signal to be reconstructed, despite the preceding and trailing 400 ms of neural data being included in the predictive model.

Feature Encoder Network: The feature encoder network is used to reduce dimensionality of the input signal before being passed to the Quantization Module and Context Network. The encoder is therefore a 1-D CNN, operating on the fixed length, single-channel, 0.5 s of 512 Hz input sEEG data. The network has 5 convolution layers, each consisting of a 1-D convolution, dropout regularization with probability $p = 0.25$, layer normalization [231], and a GELU activation function. The first convolutional layer learns 128 filters with width of 7 samples. The next two layers reduce to 64 filters with a smaller 3 sample kernel. The final two layers further reduce dimensionality to 32 filters with a kernel width of 3. All layers use no padding and a stride of 2 to reduce dimensionality. The resulting feature encoding architecture encodes a 0.5 second window of sEEG into 6 sequential steps of 32 element channel data ($32x6$).

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

The positional embedding used in brain2vec is produced from a learned transformation of the RAS coordinates described in Section 5.2.1.4. The first linear layer of the transformation receives the electrode's 3-element RAS coordinates and transforms the input to 32 hidden units. Another 32-unit hidden layer then further transforms the features, before a final output layers produces a 32 x 6 -dimensional embedding vector. A LReLU with negative slope equal to 0.01 is used as the non-linear transform after each linear layer. We use a LReLU, rather than a standard ReLU, to better handle negative values of the RAS coordinates, while still be computationally simple. The resulting embedding vector is added to the latent representation vector before being passed to the context network.

Quantization Module: The vectors are quantized using a combination of the product quantization [232] and Gumbel Softmax [233] techniques. Product quantization involves creating a set of discrete vectors by defining a number of codebooks G, each with a set of codewords W. Quantization vectors are made by concatenating codewords sampled from each codebook. Thereby a maximum number of quantization vectors is given by W^G. We assign the hyperparameters $G = 2$ and $W = 40$ for a maximum possible 1,600 vocabulary size.

Gumbel Softmax enables one-hot encoding of the quantization vectors in a fully differentiable way. A vector of $G * W = 80$ logits are provided to a Gumbel Softmax in order to produce a differentiable one-hot encoding of a word within a group. The quantization vectors are learned via a linear layer, ReLU, and another linear layer which outputs the logits. A diversity loss term, discussed in more detail in the training section, encourages diverse use of the codebook and codewords. This prevents collapse of the codebook, such that it uses only one or few codewords. Details on the exploration of the effect of modulating number of groups and words on a performance of a vector quantized approach are examined in [234].

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

This masking is the basis of the self-supervised learning of the model and is implemented according to [229]. Due to our shorter sequence dimension of only 6 elements, masking is simplified to choosing two consecutive time steps at random. Each masked latent representation is replaced by the same learnable masking token vector. Overall this results in 1/3 of latent representation vectors masked for the context network. An example of this masking is provided in Figure 36.

Context Transformer Network The context network is a transformer which follows the same architecture as the encoding side [108], also employed by BERT [13], which provides the in-depth details of the Transformer architecture. The proposed context network consists of 6 transformer block layers, each with four attention heads, 2048 feed forward units, and dropout regularization with $P = 0.25$. The output of each layer is the same dimension as the latent representations fed into the network.

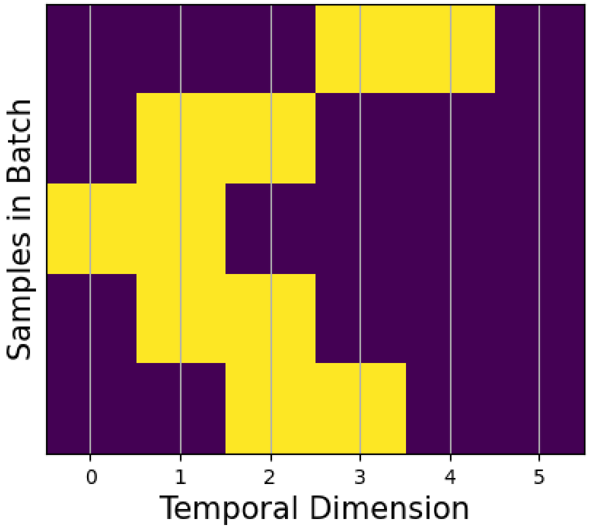


Fig. 36. Illustration of a random mask on hypothetical batch of 5 samples. A model would be required to identity the correct encoding for each of the yellow regions depicted in the figure

5.2.2.2 Pretraining

During pretraining, brain2vec learns speech activity representations from intracranial signals based on an objective function that requires it to correctly identify the true quantized latent representation vector from a set of distractors. The model only has access to the unmasked corresponding context representation vector when producing its predicted representation of the masked steps. By using discrete targets rather than continuous vector space targets, the network is influenced towards a parsimonious set of 'hidden unit' clusters which represent the underlying speech activity.

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

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

This contrastive loss is combined with a diversity loss term. The diversity loss L_d is used to ensure that the use of codewords and codebooks is diverse. The equal use of W codewords from G codebooks is encouraged by maximizing the entropy of averaged softmax distribution over the codewords for each codebook $\overline{p}_g$

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

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

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

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

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

Optimization Procedure: Models are pretrained using stochastic gradient descent, with batches of 1,024 sensor windows over 100 epochs. A random 20% of training samples, stratified at the participant-sentence level, are set aside for cross validation at the end of each epoch during training. The final model is taken from the epoch with the lowest loss L on the cross validation samples. A learning rate of 0.001 and betas of $(0.5, 0.999)$ were used with the Adam optimizer [235]. The learning rate is reduced by a factor of 0.1 every 10 epochs without improvement on a validation set drawn from the training set.

5.2.3 Evaluation on Classification Tasks

To assess the viability of brain2vec, and the generalizability of its learned representations, the features extracted through the feature encoder and context network are applied to three distinct but related downstream classification tasks. These tasks were chosen to be relevant to different aspects of speech decoding; however, they vary in complexity and the components of speech being classified. For all three classification tasks, 0.5 seconds of sEEG data from all available electrodes is considered, with labels for the half-second window assigned in a task-specific manner. In all cases, classification performance is evaluated using balanced accuracy.

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

5.2.3.1 Leave-one-participant-out Pretraining

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

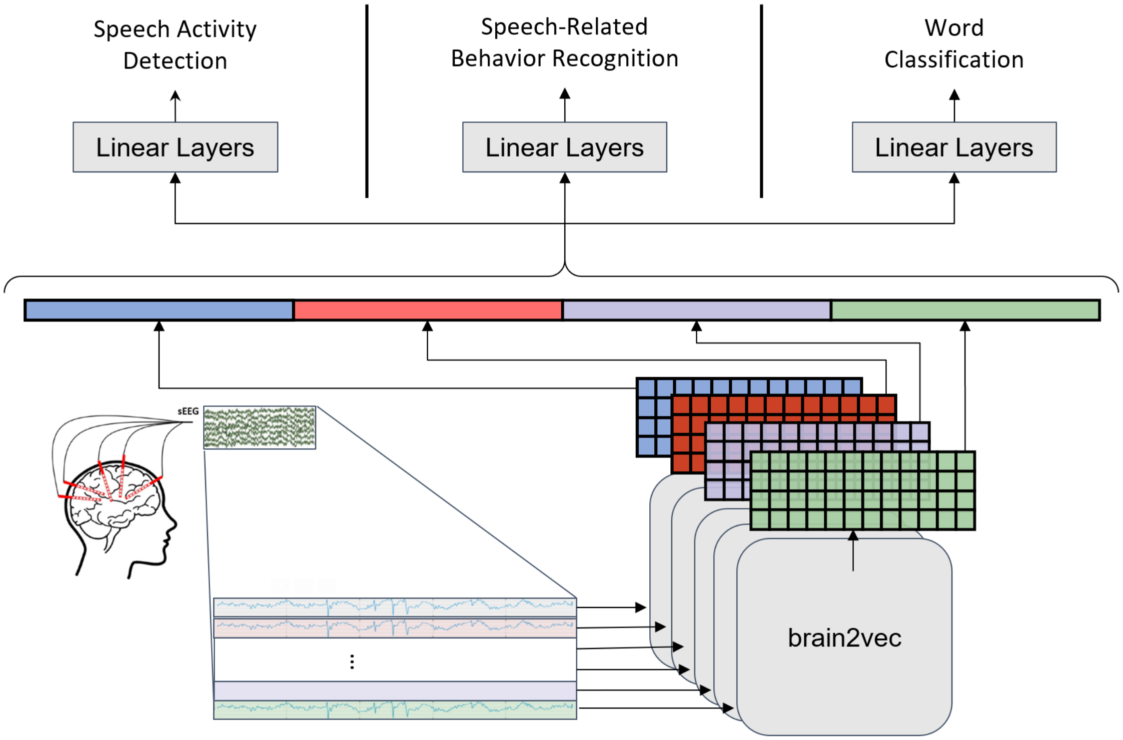


Fig. 37. Diagram of the downstream task training procedure. Given a participants Stereotactic Electroencephalography (sEEG) signals, a 0.5 s window across all electrodes is considered. The window for each single electrode, and it's corresponding Right Anterior Superior (RAS) coordinates, are passed to a brain2vec model, producing context representations for each electrode. These representations are flattened, concatenated, then passed through a 16-unit linear layer before finally being passed through the N-class classification output linear layer. The value of N-class is dependent on the task being optimized.

For each participant, that participant's data is excluded and all remaining participants' data is pooled into an unlabeled training dataset. Thus, a unique pretrained model is generated for each participant, one that has never seen a sample from the patient before fine-tuning. This paradigm minimizes data leakage in context feature learning, and ensures the model is not simply memorizing inputs. Additionally, it is intended to simulate the ultimate intended scenario for which a pretrained model based on a larger data corpus is used as the initial model for a new user and subsequently fine tuned. Herein, a pretrained brain2vec model refers to such a participant-specific, leave-one-out model. All models employ the same architecture and only differ with respect to the training data.

5.2.3.2 Downstream Classification

The utility of learned features is assessed by optimizing parsimonious supervised classification models using only the features extracted form brain2vec. The parameters of the brain2vec model are frozen, and not updated, to better assess practical applications where new data and available training time are both small. We refer to these procedures interchangeably as "fine-tuning" or "downstream classification".

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

During fine-tuning, only the additional linear layers and normalization layers are updated. The fine-tuning is performed separately for each participant. That is, a classifier is trained for each participant on their set of electrodes and corresponding labels.

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

Speech-related Behavior Recognition: The behavior recognition task labels each 0.5 s sEEG window according to one of four speech-related behaviors; *listening*, *speaking*, *mouthing*, or *imagining*. The resulting 4-class classification problem challenges the model to disambiguate highly related activities. The experiment protocol codes the regions with associated experimental cues, visualised in Figure 33. Labels are assigned to the sEEG data according to these task intervals. Each interval is 4 s in length; however, the initial 0.5 s and the final 1.0 s of the 4-s interval is not labeled to better ensure that the labeled data is representing the speech-related behavior within the interval.

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

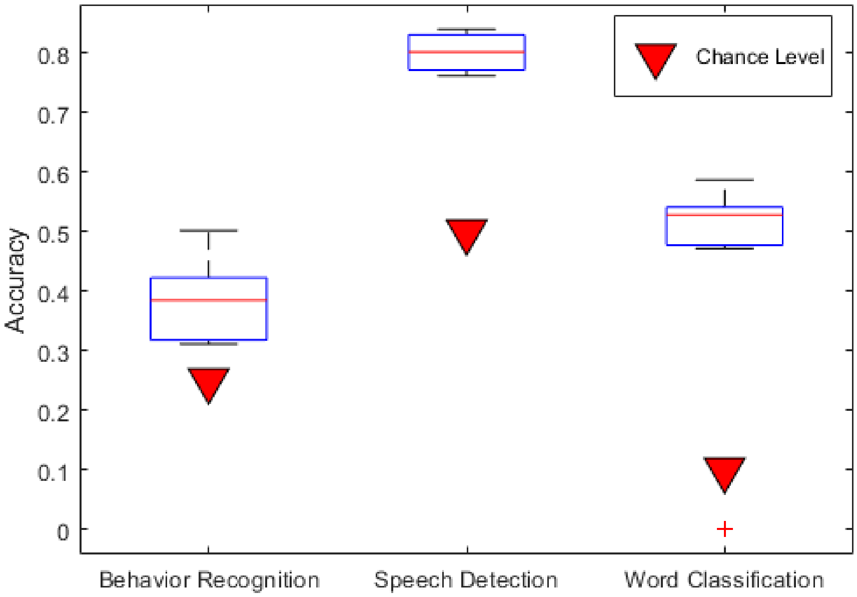


Fig. 38. Box plot of accuracy across participants for the 3 downstream task. Red triangles represent the chance accuracy for each task.

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

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

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

5.2.4 Results

The performance of brain2vec is evaluated by comparing the balanced accuracy for each of the respective classification tasks. Figure 38 and Table 5.2.4 show the balanced accuracies of the three tasks for each participant, the overall average accuracy, and the chance accuracy of the classification task. In order to verify chance accuracy, the downstream tasks were trained on randomly assigned labels, and these results are included in the table.

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

The Speech Activity Detection task's average balanced accuracy is 80.2%, and achieves the smallest variance among the tasks. All participants were significantly above chance accuracy of 50%, and the worst performer attained 82.7% accuracy. For comparison, in a recent speech activity detection study using the same Harvard Sentence dataset, logistic regression models as well as CNN models achieved an average accuracy of 82-84%[236]. Several other studies using intracranial signals reported results ranging between 80% - 94% accuracy[208, 202]. All these studies used fully supervised learning methods.

Word Classification yielded the most promising performance of the three tasks. With only one training example of each word from the repeated word set, average participant accuracy was 52.9% when tested on repeated words. Moreover, the hold-out words were from entirely different sentences with different broader context. As mentioned in Section 5.2.1.3, Participant 1 did not complete all 50 sentences during the data collection experiment. They did not have the samples required to be evaluated on the Word Classification task, and thus are excluded from this portion of the evaluation experiments.

A notable observation seen in Figure 38 is that, while there were some exceptions, there was a tendency for participants to perform consistently in comparison to other participants across the three tasks. For example, participants 4 and 6 performed in the top half for all tasks, while participant 3 and 7 performed in the bottom half.

Figure 39 shows the cross-validation loss of during pretraining for all participants. It can be observed that the models converge to generally similar losses, that is, there do not appear to be order-of-magnitude differences. This is expected, as each model shares approximately 6/7 of the electrode data corpus. Nevertheless, it is confirmation of that there is some measure of consistency in the convergence process.

The confusion matrices of downstream classification tasks are shown in Figure 40. The Behavior Recognition task shows that *imagining* was confused more often with *listening* and *mouthing* than with *speaking*. Further, *speaking* was confused most often with *mouthing*. This observation may indicate a closer mechanistic relationship between imagined speech and listening or mouthing than over speaking [237, 238, 239].

Figures 41, 42, and 43, and respectively show the 3-component t-SNE [240] of the pretrained features for each fine-tuning task. The figures give an indication that the context representations learned by brain2vec are meaningful to each speech domain task. It is observed that, for each task, there are clear regions of separability for each of the classes. Particularly, word classification in Figure 41 shows distinct differentiations between words. This likely contributes to the impressive performance of the word classification task given comparatively little training data, as the context representations show clear differentiation prior to supervised training.

5.2.5 Related Work

There have been significant advances in neural speech decoding over the past decade using intracranial recordings such as ECoG or sEEG. These include describing brain regions and mechanisms involved in speech, predicting words or phonemes, translating neural signals to articulatory kinematics models, text, or directly to speech waveforms [195, 189, 196, 197, 198, 199, 200]. Recent efforts have progressed to real-time synthesis or classification, and decoding of imagined speech [192, 201, 202, 203, 204, 205].

However, due to the nature and limitations of the clinical procedures commonly used to obtain research data, existing methods for neural speech decoding generally rely on participant-specific models, trained on labeled experiment tasks. Supervised approaches such as these are

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

Table 13. Balanced accuracy of downstream tasks. Participant 1 did not have a complete dataset needed for Word Classification and is therefore omitted.

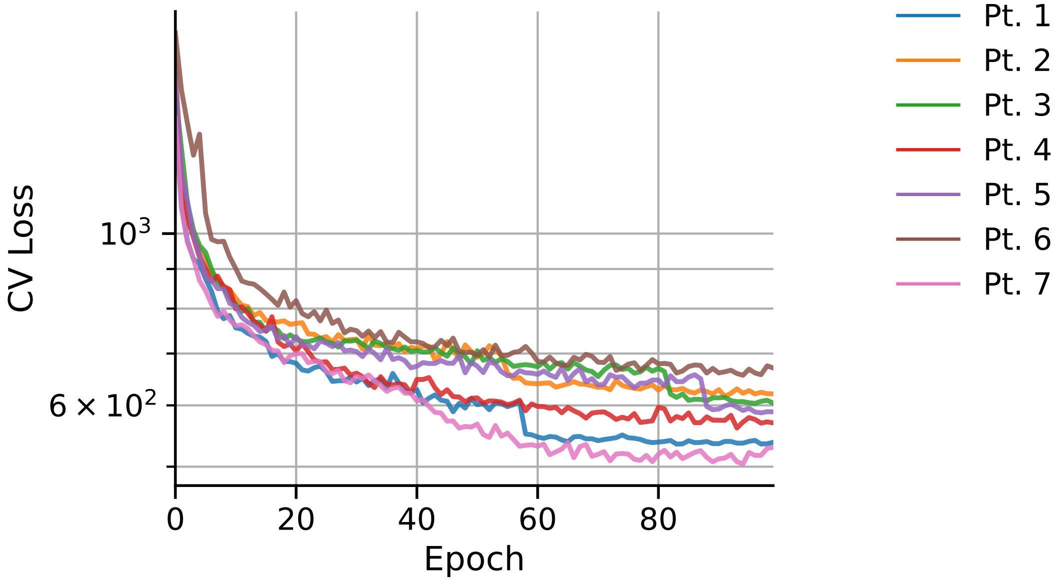


Fig. 39. Cross validation loss of brain2vec model over pretraining epochs.

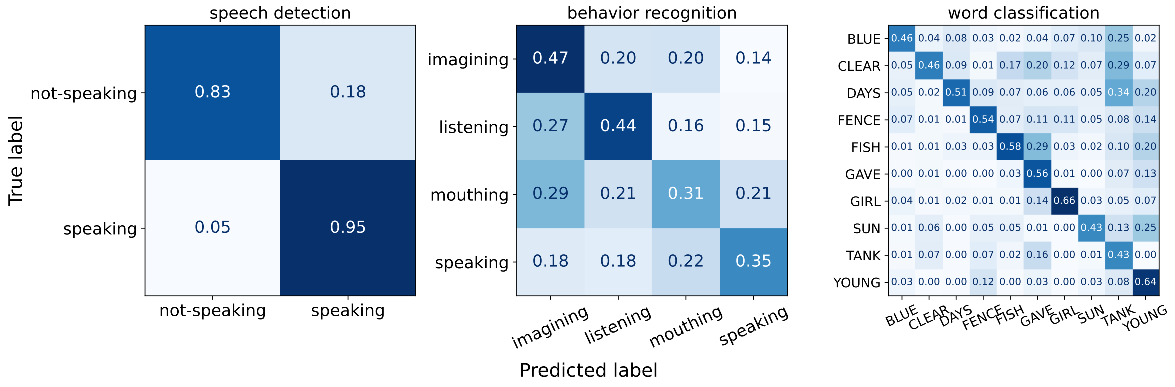


Fig. 40. Confusion matrices of fine-tuning classification tasks across all participant test sets. Each row (true label) is normalized independently, giving the portion predicted class labels across all of the true samples evaluated.

naturally restrictive, supporting only one particular participant's sensor configuration and task-related behavior. Instead, SSL methods with unlabeled data and explicit handling of sensor configuration may allow for much more flexible paradigms in which multiple participant's data can be pooled for learning general purpose features. Furthermore, methods that learn without labels have broader potential applications, including use in closed-loop online systems in which labels are unreliable or non-existent.

SSL enables optimization of adaptable representations using only unlabeled data [241, 242, 243, 244, 245]. Methods *pretrain* a model on *contexts* implicit within the data. These *pretexts* typically exploit assumptions of locality to encourage information-rich representations relevant to downstream modeling tasks. There are many approaches to pretraining, mostly as variations of generative and contrastive objective terms [246], with the pioneering success in NLP [247, 248] and recent advances in visual domains [249]. Model architectures vary depending on the domain and modality, but the *transformer* [108] is often an architecture component for SSL approaches.

The recent introduction of the transformer architecture ushered in a new era for the deep learning field, showing the attention mechanism to be a simple yet powerful tool for NLP and sequence to sequence models [108]. The self-attention transformer block served as the foundation for BERT [13] and the GPT series [14], which solidified a trend of self-supervised learning where models are pretrained on a large, neutral, data corpus before being fine-tuned on a specific task of narrower scope. More recent vision transformers effectively demonstrate that most data can be treated as a sequence, that self-attention performs as well or better than CNNs, and that computer vision models can benefit from self-supervised pretraining like their NLP counterparts [250]. Transformers have since been shown a viable or superior method for object detection, video action recognition, point cloud shape classification, and multi-modal models [101, 102, 103, 104, 105, 106].

Recently, several studies have explored training language models directly from audio signals rather than text [229, 251, 252]. The key insight of these methods is that, rather than learning a representation in a latent space with continuous targets, they learn from a discretized set of

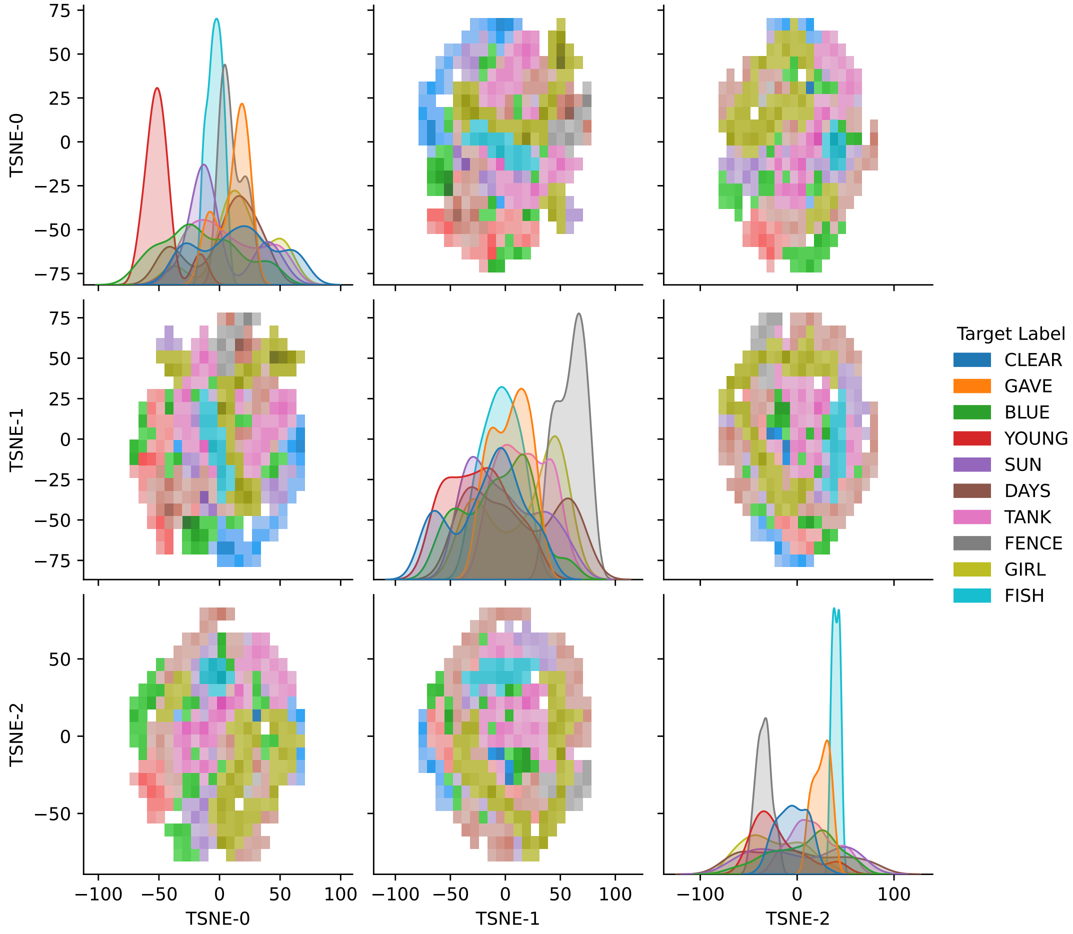


Fig. 41. Visualization of 3 t-SNE components from the pretrained features on an unseen users data (Pt. 7), colored by the **Word Classification** fine-tuning task.

'pseudo-speech' units. Thus, these methods essentially use clustering to learn a self-defined lexicon rather than being constrained to map to an externally defined set such as words, phonemes, or characters. This approach is particularly appealing to speech neuroprosthetic development because it is analogous to the way speech is processed by humans, assigning discrete conceptual meaning to physiological inputs from a persisting audio source, which are also concepts underlying speech production.

5.2.6 Discussion

The performance of brain2vec on the three disparate downstream tasks showcases the generalizabilty of the self-supervised features learned by the procedure. While all tasks achieve better than chance accuracy for all participants, in particular, the speech detection task approaches accuracies on par with other supervised learning methods, and the word classification task exhibits promising results using only a small amount of labeled data.

The main objective of this analysis was to develop and establish the efficacy of the pretraining procedure and model, using the performance on downstream tasks as a measure rather than an end goal. The manner in which the model pretrains inherently makes it difficult to draw conclusions directly from analyzing the context representations, and is further complicated with

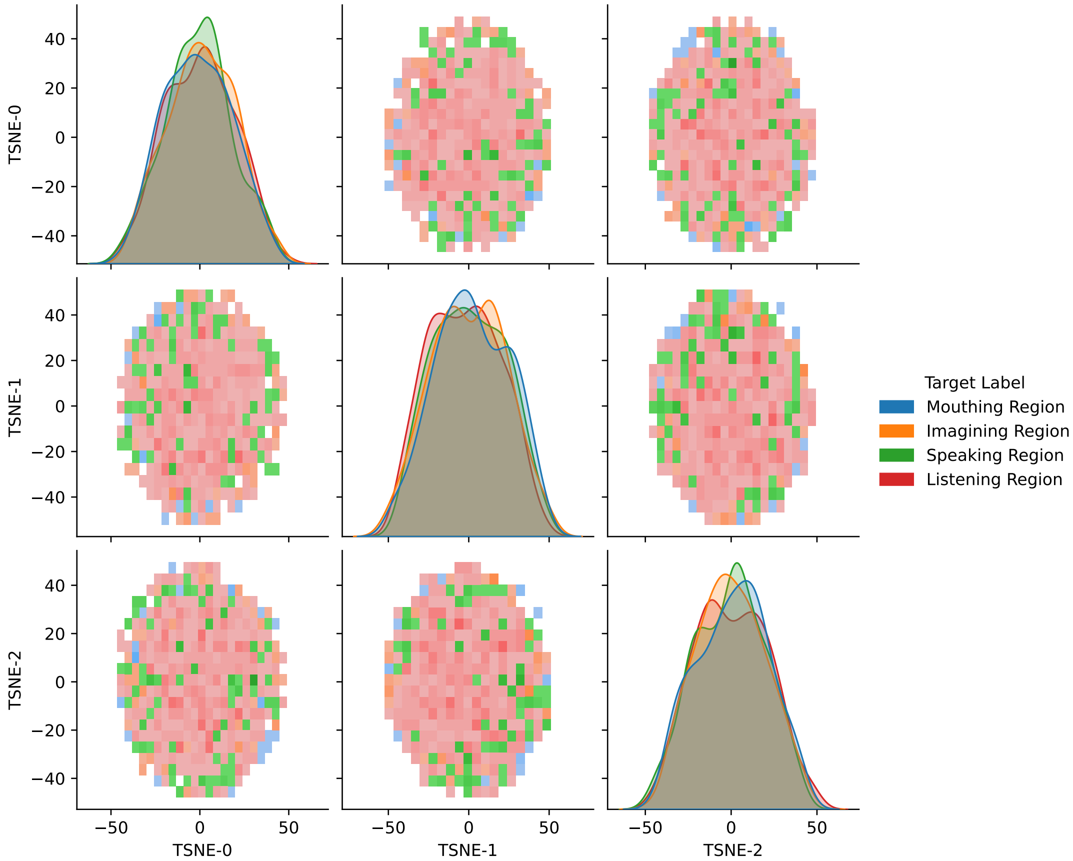


Fig. 42. Visualization of 3 t-SNE components from the pretrained features on an unseen users data (Pt. 7), colored by the **Behavior Recognition** fine-tuning task.

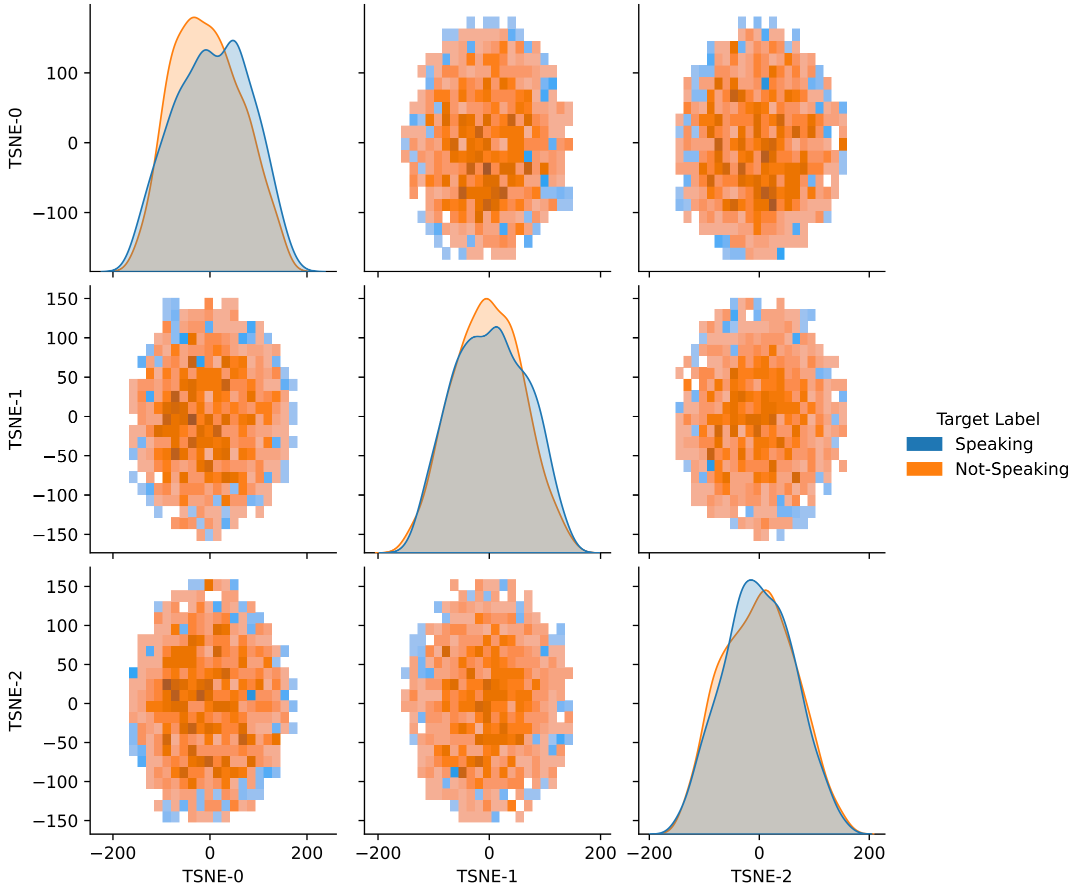


Fig. 43. Visualization of 3 t-SNE components from the pretrained features on an unseen users data (Pt. 7), colored by the **Speech Detection** fine-tuning task.

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

Performance on the Speech-related Behavior Recognition task, while comparatively exhibiting the weakest performance, can also be considered the most challenging of the three classification tasks. The neural circuits for perceiving speech, and producing overt, mouthed, and imagined speech, are highly intertwined [253, 238, 254]. Nevertheless, it is encouraging that the context representations of the model appear to encode some neural correlates of these behaviors.

The Word Classification task is essentially a few-shot learner, only provided a pair of training examples (i.e., word utterances) of each class before evaluation - one for optimization, and another for validation. In contrast, a study recently showed results ranging from 30-60% on a similar classification task using ECoG signals and a transformer architecture, though in a fully supervised manner[255]. This demonstrates the utility of the self-supervised method: using only unlabeled data, features are learned and guided into hidden, likely sub-word, units. Then, it is posited, comparatively little data is required to map these features to a word space.

The success of brain2vec is likely due to several factors. The self-supervised training of latent representations with quantized targets, while keeping the learned context representation as continuous, is a gentle influence to learn not fully-discrete codewords, but instead grouped clusters in the continuous space, known as hidden units. In this way, features are guided towards self-determined clusters, while still allowing the model to fully leverage the rich context of continuous-space features. Because of the self-supervised nature, these clusters are not matched to any linguistic unit, such as words or phonemes, and instead are self-determined by the network. However, because the training data are strictly from the speech domain, it is likely that the hidden units are converging to neural versions of some, possibly combinations of, linguistic units. This is a potential explanation as to why the Word Classification task was successful using sparse training data.

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

During model development, several issues were observed that adversely impacted training success. The objective term weights, α and λ, required exploration with small experiments to find appropriate configurations that avoided codebook collapse - wherein the model used few codewords or the codewords would have little variance overall. Under some conditions, brain2vec would fail to converge and maintained at a high CV loss, but this could not be consistently replicated and never occurred with the configuration presented in this work. We found large improvements in consistency after implementing appropriate weight initialization. Convolution and linear layers were initialized from $\mathcal{N}(0, 0.02)$, BatchNorm parameters from $\mathcal{N}(1., 0.02)$ with a bias of zero, and LayerNorm parameters are initialized with 1.0 and zero bias. This implies a sensitivity to initial conditions and hints at further improvement through more sophisticated initialization schemes and complex learning rate paradigms as explored in other language model methods [252, 234]. This is likely an attribute of the model architecture rather than the particular data.

The number of transformer blocks, and the latent representation vector dimension, and other factors that determined model complexity, often impact performance on downstream tasks. This is likely a balance with the amount of available data. Language models using transformer architectures often have a 'large' model variant with 24 transformer blocks [229, 251, 252]; however, these models are typically pretrained using on the order of 60,000 hours of data, whereas the proposed approach was effective using slightly over 1 hour of data for pretraining.

Additional sEEG training data would allow for a deeper model with more transformer blocks, a longer input sequence, or larger embedding dimension, which might in turn provide greater context and learn richer representations of multiple speech and speech related processes. The downstream tasks explored here are constrained by the nature of the speech data available. With enough data, and a sufficient depth of network, it is conceivable for brain2vec to serve as the backbone of an even more generalized model; one capable of discriminating overt or imagined speech intention, then decoding the speech from the same initial feature set.

5.3 Privacy and Performance of Neural Speech Representations

In Section 5.2, we contributed a self-supervised neural speech representation that enables transfer learning to new contexts by pretraining on multiple individuals' data. However, as discussed in Section 2.1.3, models built from potentially sensitive data, such as neural recordings, risk leaking information and facilitating downstream privacy violations of the pretraining cohort. In order to assess these risks, and improve understanding of neural speech representations, this section examines the *brain2vec* model's pretraining process and evaluates two threat models for pretraining membership privacy. We use various combinations of the seven participants' recordings to evaluate different dataset sizes and privacy issues related to pretraining brain2vec. The sequence of experiments we perform in this section are illustrated in Figure 44. Experiments use participant data from the Harvard Sentences dataset described earlier in Section 5.2.

We first characterize the impact of changes to pretraining hyperparameters when using single-participant datasets (1-participant). Experiments explore how the pretraining losses are impacted by the feature extraction architecture, positional enbedding method, codebook size, and mask length. From the results of the single-participant experiments, we select several hyperparameter configurations for pretraining on larger six-participant cohorts (6-particpant). We compare the selected models on the fine-tuning tasks originally presented in Section 5.2. Through the variations in dataset size and hyperparameter values, we begin to assess which aspects, if any, lead to better downstream classification results.

We also compare the selected model's behavior within two privacy evaluation experiments: **re-identification** and **membership inference**. The **re-identification experiment** is designed to demonstrate the feasibility of discriminating between individual participants using the representations learned from pretraining. Our approach reuses the 6-participant models to optimize 7-class re-identification models - one class for each of 7 participants, including the pretraining holdout participant. The **membership inference experiment** uses a novel two-participant (2-participant) shadow modeling design to demonstrate the feasibility of an adversary determining if a sample belongs to an individual in the pretraining cohort. Our approach requires models that are pretrained on only two participants in order to simulate private target datasets and an adversary's shadow modeling datasets. Our methods investigate both the potential for neural representations to improve generalizeability of brain-computer interfaces as well as the potential for privacy attacks against people contributing pretraining data.

The remainder of this section is organized as follows. In Section 5.3.1 we summarize the data collection and its distribution discrepancies. In Section 5.3.2 we describe our approach to experimenting with brain2vec's architecture configuration. In Section 5.3.3 we describe our methods for evaluating the privacy of individuals contributing to brain2vec's pretraining process. In Section 5.3.4 we present the results of our methods, followed by further discussion of related work in 5.3.5 and our contributions in 5.3.6.

5.3.1 Dataset & Distribution Discrepancies

The contributions in Section 5.3 use the same dataset first described as Section 5.2: sEEG data collected from seven native English-speaking participants being monitored for treatment of in-tractable epilepsy at University of California San Diego (UCSD) Health. Table 14 provides an overview of each participants' data. This section highlights the data collection protocol and the discrepancies in the data distributions across participants that might influence experiments - complete details of the data collection can be found in Section 5.2.1.

The data collection protocol for the neural data we use, illustrated in Figure 33, is designed

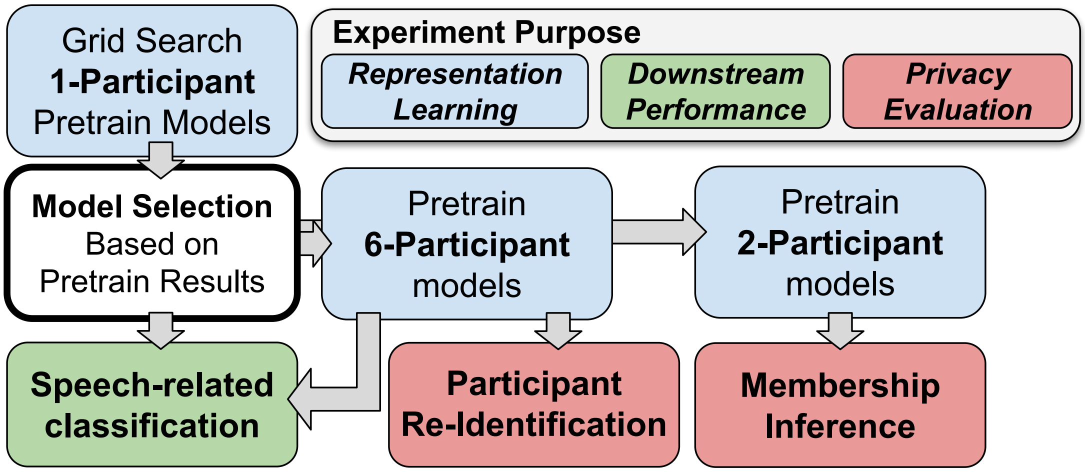


Fig. 44. Experiments and model selection workflow for evaluating brain2vecs's neural representations' hyperparameters, potential for information leakage, as well as its utility on varied pretraining dataset sizes.

to investigate both imagined and overt speech modalities. During a trial, study participants are asked to listen to a displayed sentence, then they are prompted to first repeat the sentence aloud, then only articulate the sentence without vocalizing (i.e., "mouthing"), and finally only imagine speaking the sentence without motor function. Each modality is performed by the participant in a series of 4 second intervals, each separated by a 20 millisecond blank screen. Participants are made aware of the protocol, including icons and cues, and asked to perform the associated task immediately after a cue is presented, but within the 4 second interval. Each of seven participants repeated the experimental protocol for 50 distinct sentences designed to be phonetically balanced conversational English [256]. However, the labeling was corrupted for 25 of the 50 trials recorded for Participant 1, leaving only 25 sentences useable for our research.

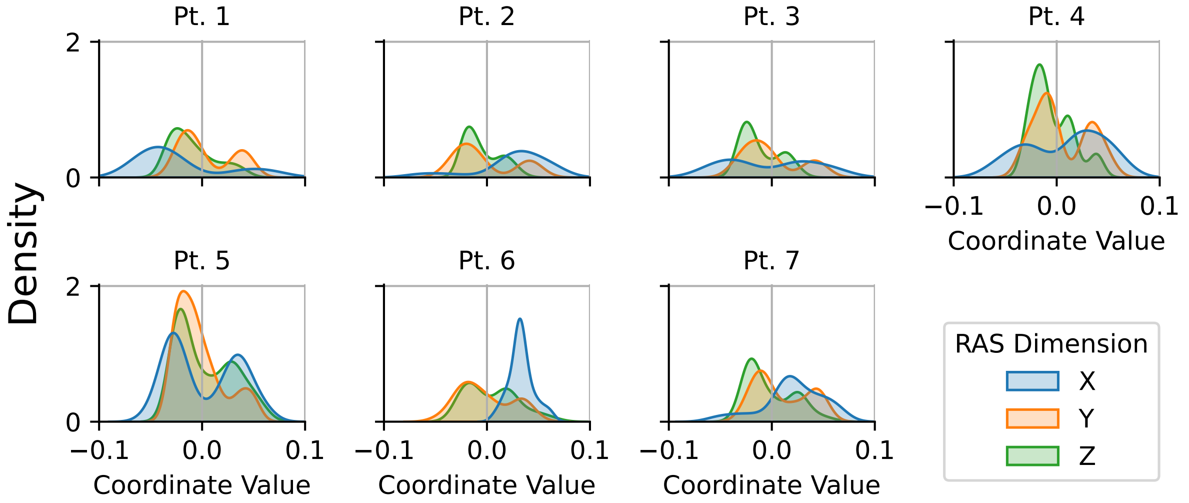


Fig. 45. Distribution of coordinates in a shared RAS coordinate frame across participants.

Table 14. Summary of Participant sEEG Recordings

Participant ID	**Num. Sensors**	**Duration (Min:Sec)**	**Num. Samples**	**Avgerage**	**Standard Dev.**
1	90	17:05	47,250,000	5.2	61.8
2	70	13:50	29,750,070	4.9	80.1
3	80	14:48	36,400,000	-2.4	78.2
4	175	14:38	78,750,000	2.5	181.1
5	232	18:52	134,560,000	-1.0	102.8
6	94	15:57	46,060,000	2.7	117.4
7	108	14:38	48,600,108	0.2	84.5

The location of each sensor in the brain is important for contextualizing the neural signals being recorded. We use the same *volumetric morphing of electrode locations to a common brain atlas* that is described in Section 5.2 to preprocess sensor coordinates. The resulting (X, Y, Z) coordinates are in meters with positive values corresponding to right (vs. left), anterior (vs. posterior), and superior (vs. inferior). These data are referred to as RAS coordinates and their distribution across participants is visualized in Figure 45. The RAS coordinates are used in the original design as the inputs to a dense network that learns a positional embedding for the context encoder.

Distribution discrepancy is a key challenge in BCI and other HAR problems - training data may simply not match evaluation or application data, and sufficient data for analyses can be challenging to collect given the diversity and costs. Our dataset for this work is no different. There is considerable spatial discrepancy across participants' electrode location as measured by our RAS coordinate's. For example, participant 6's electrodes are located entirely in the right hemisphere, while the other participants, such as 3, 4, and 5, are distributed more evenly across both hemispheres of the brain. Our experiments in later sections will explore performance when a model is only trained on a single participant's RAS distribution, and transferred to a different participant. Using RAS to encode the position of the sensor data may therefore be critical to determining downstream model behavior. Separately, we visualize the spectral distribution of participants' sensors. We apply an FFT to all sensors, across the entire dataset, for each participant and visualize the resulting distribution in our Background chapter's Figure 7. Each participant's spectra follows an expected pink-noise distribution, but large variations in magnitude also exists across frequencies. These spatial and frequency discrepancies across participants are expected to challenge models in the following experiments in which we train models that have only seen one or two participants before being applied to all other participants.

5.3.2 Grid Search of Pretraining Configurations

We characterize our neural representation learning methodology by optimizing models over various cohort sizes and model configurations. We first aim to understand brain2vec pretraining on individual participants and the impact of varying codebook size, mask length, positional embedding method, and feature extractor architecture. We use individual participants to provide insight into how the method may perform differently when only applied to a single sensor configuration (represented through the RAS coordinates), sensor value distribution, and reduced dataset size. An added benefit is that the reduced dataset size speeds up pretraining, allowing a broader search of configurations. From the results of pretraining each hyperparameter combination, we select the best performing configurations for each positional embedding method. We don't select the positional embedding method based on 1-participant results since performance may be impacted by varied RAS coordinates and sensor values across more participants. We also perform additional optimization across context network depth using the best 1-participant model configurations. We then take the best performing context network depth on 1-participant datasets and apply those pretrained models to the original downstream fine-tuning tasks. We also use these configurations to pretrain 6-participant models and examine their performance. Finally, we also use these configurations in our privacy experiments introduced in Section 5.3.3.

The set of hyperparameter configurations, whose unique combinations we optimize across all 7 individual participants as 1-participant models, are provide in Table 15. Below we summarize the motivation for each hyperparameter modification, with additional description for new methodolgies. We conclude this section with a description of the pretraining methods that, unless otherwise noted, is used throughout Section 5.3.

Hyperparameters	Configurations
Feature Extractor Architecture (N Filters x Width x Stride)	128x7x2|64x3x2|64x3x2|32x3x2|32x3x2 = Shape(6, 32) 128x7x7|64x5x5|16x3x2 = Shape(3, 16)
Positional Embedding Method	Spatio-temporal from RAS Spatial from RAS
Codebook Size	20, 40, 80
Mask Length	1, 2

Table 15. Hyperparameter configurations used in single participant pretraining experiments. Description of each configuration is provided in Section 5.3.2

Feature Extractor Architecture: We experiment with the impact of the feature extractor by reducing its depth, while increasing the width and stride to reduce the embedding dimension to 16 elements and temporal width to 3 steps. We refer to this configuration as **16x3** and the original configuration as **32x6**. The smaller 16x3 model speeds up training and reduces the size of the resulting extracted features, improving it's potential utility in practical applications. We hypothesize that the reduced dimensionality may also help prevent over-fitting in downstream fine-tuning tasks. However, the reduced dimensionality of the feature extractor's output will also impact the quantization. Lower dimensional features restrict the quantization, making it more challenging to utilize larger codebooks. This may bias models to learn simplified lexicons, no matter how many quantization words and groups are used in the codebook.

Positional Embedding Method: We also adjust how the positional embedding is learned and experiment with an implementation that does not embed across time from the RAS coordinates. The original brain2vec described earlier in Section 5.2 learns a spatio-temporal embedding directly from the RAS coordinates. As described in 5.2, the RAS coordinates are passed through a multi-layer linear network with LReLU activations to produce an output size equal to the embedding and time dimensions (e.g., $32x6$ in the original model). However, we hypothesize that this implementation may overfit to the RAS dimensions of the pretraining cohort and transfer poorly to new participants and tasks. Intuitively, it may be more helpful to decouple the representations of time from the representation of space. Therefore, we also experiment with a positional embedding that only encodes the spatial dimension from the RAS coordinates and directly learns a positional embedding for each time-step with the same width as the embedding dimension. This method intends to disentangle the spatial representation from the temporal representation. We hypothesize that his modification may help the model transfer across different participants' spatial configurations.

Codebook Size: We vary the codebook size W, with the number of codebooks fixed to $G = 2$. Increasing the codebook size allows the model to be more expressive and varied with it's representations. However, it's also possible that reducing codebook size may help regularize and prevent over-fitting in downstream fine-tuning tasks. Therefore, we experiment with the original configuration of $W = 40$, as well as halving the value to $W = 20$ and doubling the value to $W = 80$.

Mask Length: We experiment with halving the original mask length configuration of 2 to a mask length 1. Increasing the mask length to 3 or more is not possible with our smaller 16x3 feature extractor's temporal size of 3 steps. We perform these experiments because it's unclear how this configuration change might impact the pretraining process and the resulting learned representations. Masking more of the latent features will make the pretraining task more difficult, impacting the contrastive loss and possibly the utilization of the codebook as captured through the diversity loss.

Pretraining method: Given our limited data, we only use the first 40% of samples for pretraining to avoid *over-fitting* to the dataset since downstream fine-tuning tasks and privacy experiments need unseen samples for evaluation. This aligns with guidance provided in prior work for evaluating SSL algorithms like brain2vec [257]. We fix the number of codebook groups $G = 2$ to limit the search space while still enabling the non-linearity of multiple codebooks. We also keep fixed the assignments of the learning rate $l = 0.001$, the diversity weight $\alpha = 1$, and the feature penalization weight $\lambda = 10^{-1}$. We use a batch size of 1024 samples. All models are optimized with early stopping configured for 15 epochs and learning rate reduction by a factor of 0.1 every 10 epochs without improvement. Because we limit pretraining data and stop models early, we expect these experiments to result in reduced performance overall when compared to the previous chapter's results. Still, we expect that our experiments can still illicit important variance across model configurations, even with these additional restrictions.

5.3.3 Privacy Evaluation Experiments

We perform two experiments to assess the degree of information leakage resulting from brain2vec and its pretraining methodology. Both methods are focused on the privacy of an individual participant who has contributed data to the pretraining of a brain2vec model. Our contributions assume a desire to measure and possibly reduce the risk of privacy violation in support of trustworthy AI for bioelectric HAR. In the context of our BCI application, this means that an individual's decision to contribute neural data to the construction of a shared embedding model must not imply a forfeiture of their confidentiality. Any risks to potential misuses or unintended information leakage should be measure and appropriately communicated to participants.

We use brain2vec because it is our own contribution and because it is able to generalize across multiple participants' unlabeled data, laying the foundation for many users to contribute data to build a more robust model. Such a model may then be shared broadly to quickly enable new individuals and their BCI solutions. In such applications, measuring the risk of privacy violation is valuable since it may improve trust, regardless of the findings. We frame our privacy experiments by describing the assumptions surrounding a potential attack, known generally as a *threat model*. Therefore, these experiments may also be considered simulation of privacy attacks against brain2vec models by some potential *adversary* or *attacker*. The model instance being attacked is referred to as the *target* model. We first establish shared notation for our privacy attacks, followed by a detailed description of each experiments' methods and practical motivations.

Notation: In our ML-based paradigm, a model M trains using algorithm A on dataset D, which is made up of individual users $u_i \in U$. The algorithm A represents brain2vec's pretraining methodology, which remains fixed except for the hyperparameter variations selected from experiments described previously in Section 5.3.1. We vary the dataset D with distinct combinations of users to produce different *target* models M_{target} built using algorithm A for training. A simulated adversary attempts to predict which individual produced a sample of new data (re-identification) or predict whether the user was originally part of target model's training dataset D_{target} (membership inference). In each case, the adversary intends to use M_{target}'s outputs to perform the attack and violate the user's privacy. The following methods for privacy evaluation share these basic concepts, but the details of each experiment's design are provided in the next subsections.

5.3.3.1 Participant Re-Identification Task

Sharing an individual's data in order to create a larger database of important attributes and outcomes is foundational to modern health science and ML research. However, a core concern in sharing private sensitive data is the ability to re-identify the individual tied to the data in order to violate their privacy. We discuss the history of privacy and how it relates to contemporary methods in our background chapter's Section 2.1.3. We build on prior research and measure brain2vec's privacy risk as the likelihood that a sample can be linked to the individual who provided the data. We make generous assumptions of an attackers access to the pretraining data in order to perform an assessment of re-identification attack feasibility. Our experimental methods are described in this section with results provided later in Section 5.3.4.

The re-identification attack is designed similar to a fine-tuning task, but one that seeks to identify the individual with only the pretrained model and samples of new data from the participants being identified. The feature extraction process learned by brain2vec is general-purpose - the downstream task may conceptually be any approach that maps sensor readings to some other target distribution. Our contribution assumes the null hypothesis that brain2vec's learned representations are not correlated with the participants' identities, but only correlate with participant-agnostic downstream tasks. Our re-identification experiments test this assumption, examining if we can build a model to re-identify the pretraining participants. In practical scenarios, our re-identification experiment simulates a threat models in which a malicious model developer with access to the pretraining dataset desires to re-identify their pretraining participants from unlabeled neural data in future applications. Importantly, in applications such as security, there is a practical need to be able to re-identify a previously authorized individual. Even entertainment-oriented use-cases of BCI (e.g., interactive simulations or video games) may desire to re-identify a user in order to load their preferences, for example. Therefore, a successful re-identification attack highlights both the utility of brain2vec as a biometric authentication tool as well as its potential for use in violating an individual's privacy. In contrast, a failure to re-identify participants suggests that brain2vec cannot be used as a biometric authentication method and that raw feature outputs are poor attack vectors for re-identification.

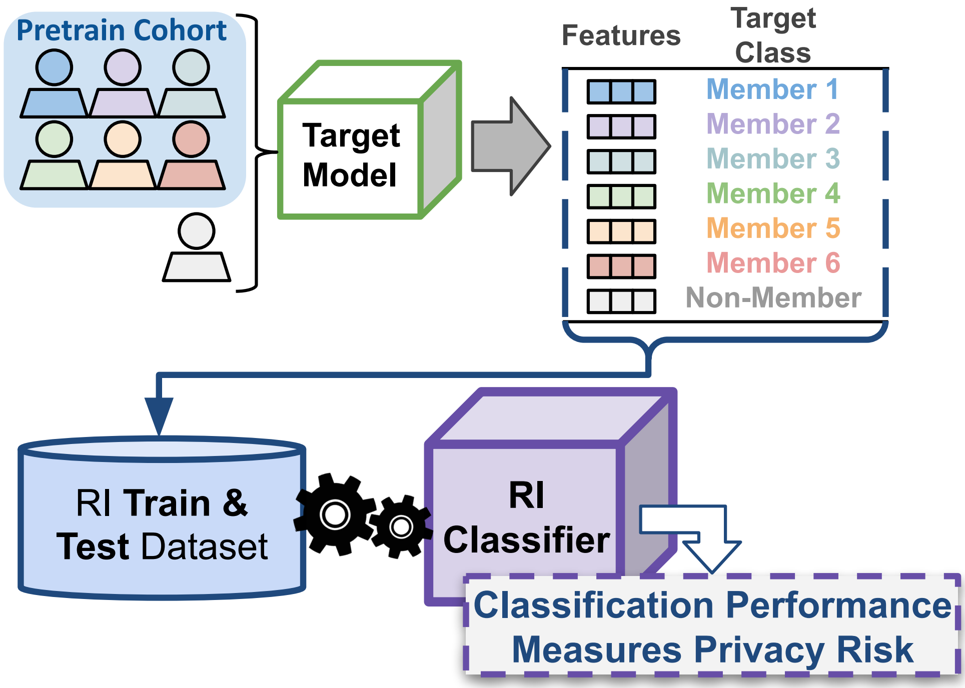


Fig. 46. **Method to measure risk of participant Re-identification:** A pretraining cohort of 6 participants is used to optimize a brain2vec model. The adversary uses their access to the training data to build a re-identification classifier training set from the 6 pretraining participants and a remaining participant not part of the pretraining cohort. The model is evaluated using the same participants data, only we partition the dataset and use the last 40% of samples, which were not used during pretraining.

Methodology: A private target model M_{target} is pretrained on a set D_{target} that is made up of neural data from a set of participants U_{target}. Suppose an adversary with access to M_{target} wishes to also build a *re-identification* classifier C_{RI} that can identify which individual produced a sample of neural data using only M_{target} for feature extraction. Importantly, the pretraining process for M_{target} is not modified to support the needs of C_{RI}. Instead, M_{target} is pretrained for general-purpose downstream tasks as previously outlined. We assume *black-box* access to M_{target}'s outputs - C_{RI} can only access the outputs of M_{target} and cannot access potentially useful hidden features or M_{target}'s inputs. This enables attacks to be used in scenarios where access to the model's inputs or parameters is forbidden or obfuscated. Our assumptions of black-box access together with our assumption that training C_{RI} does not modify M_{target} also simulates broad scenarios in which the training algorithm A may not be known by the attacker when building C_{RI}.

Our re-identification experiment is illustrated in Figure 46 and uses all 7 participants to evaluate privacy of the 6-participant brain2vec models selected from the results of Section 5.3.1.

In each experiment, one individual does not contribute to M_{target}'s pretraining, allowing us to compare privacy risks for pretraining members vs. non-members. M_{target} is trained using the methods described in Section 5.3.2, therefore M_{target} is only pretrained on the first 40% of samples from the participants. Similarly, during training of C_{RI}, data is restricted to only the same first portion of samples in each of users' trial. We reserve the remaining portion for drawing test samples. Taking test samples from the later portion of the experiment trial is intended to simulate an attack using the re-identification model sometime in the future after training on the original pretraining data D_{target}. This design is intended to further simulate practical scenarios in which time passes between C_{RI}'s development and its use in future re-identification.

The re-identification attack requires that the attacking model (i.e., C_{RI}) be applied to multiple users, even when individuals have varying numbers of sensors. To approach this issue, we make a simplifying assumption that also increases the difficulty for the attacker: we assume that the attacker will only have access to features extracted from a random set of N_s sensors. The attacker knows they will receive features for N_s features, but they do not know which sensors. This approach is implemented by sampling, without replacement, N_s sensors from each participant's window of data before being combined with other participant's data within a batch. Sensors are selected based on a uniform distribution - no particular sensor is preferred over any others. We then experiment with re-identification attacks that vary N_s over each model configuration. This allows our experiments to gauge how increased sensor access might increase the risk of re-identification. Note that we only evaluate up to $N_s = 64$ randomly sampled sensors since one the participants in our dataset only has 70 sensors.

The C_{RI} model is a fully connected feed-forward neural network with 2 hidden layers having 128 units and a final layer with 7 output units. LReLU with a negative slope of 0.01 is used for hidden activation, with a softmax output for classification. We use a element-wise dropout rate of .75 and batch normalization to regularize the model. Otherwise, we use the same training configurations previously described. We do not search the hyperparameter space of C_{RI}, therefore it is intended to be simple, yet expressive enough to confidently assess privacy risk. Results for the re-identification experiments are provided in Section 5.3.4.

5.3.3.2 Membership Inference

When attempting to attack an individual's privacy using public ML models, an adversary may try to determine whether an individual was a member of the a model's training cohort. This attack is known as *membership inference*. If an attacker is able to infer membership, then it follows that they can also tie other attributes of the dataset to that individual. For example, inferring that an individual was a member of a dataset used to train a model on a specific diseases prognosis also confers to an attacker the knowledge that the individual has the disease. In this section, we describe our method of simulating a membership inference attack against brain2vec. Our experiments aim to infer the individual's membership in the training data, as opposed to specific samples from the individual. In other words, given an individual who provided neural data for pretraining, any sample of neural data from that individual is considered a member of the pretrained model's cohort. Our experimental methods are described in this section with results provided later in Section 5.3.4.

Similar to our re-identification attack, we assume that brain2vec's pretraining procedure is intended to produce features uncorrelated with the individual contributing the input data. Our membership inference experiments therefore examine if the feature distributions are altered when an individual that wasn't seen during pretraining provides input. It follows that in order for membership inference to be successful, an attack model must be able to discover a generalizeable mapping from the feature distribution to the membership class. Stated differently - the way in which brain2vec's output distributions are altered when applied to new participants must be reproducible across model builds. We expect this to be a difficult problem, primarily due to the self-supervised nature of brain2vec - it's features are learned with minimal guidance, and the change in distribution between members and non-members is likely highly divergent between model instances. As discussed in Section 5.2, the benefit of the brain2vec methodology is its ability to discover its own lexicon for representing neural signals. This lexicon, let alone how it changes between participants, has no constraints that would encourage its encoding of the representation to remain consistent across separate model optimizations. For this reason, we experiment with a white-box attack in which the attacker can access hidden states and the self-supervised loss outputs for a sample. The self-supervised losses measure the ability for the model to infer masked steps in the encoded features, and we hypothesize that this correlates with pretraining membership. The losses measure a sample's usage of a model's codebook as well as the context network's ability to produce similar feature representations. Utilizing the pretraining loss information as a vector for membership attack can be conceptualized as asking the model *"how well can you encode this sample of data?"* - we expect the response may be indicative of pretraining membership. A white-box threat model such as this is a generous assumption for the attacker since real-world applications may restrict this access, but we choose this approach in order to set a baseline for what we hypothesize is a difficult attack.

Methodology: A private target model M_{target} is pretrained on a dataset D_{target} that includes data from participants in U_{target}. An adversary desires to infer if an individual u_i contributed data to the target model M_{target}, meaning they wish to infer if $u_i \in U_{target}$. The adversary knows the pretraining algorithm A and the number of individuals $|U_{target}|$ used to pretrain M_{target}. The adversary also has access to another set of individuals U_{adv} who can contribute data for the adversary's needs. The users in U_{adv} are not members of U_{target} - U_{adv} and U_{target} are disjoint sets of users. The adversary uses U_{adv} to construct a shadow modeling dataset D_{sm} to train a membership classification model C_{MI}. The goal of C_{MI} is to predict if a participant u_i is a member of that model's users U using outputs provided by the model's interface. Therefore, the adversary uses U_{adv} to simulate various training cohorts and membership combinations for training C_{MI}. Specifically, for every unique combination of $n = |D_{target}|$ participants in U_{adv}, the adversary trains a shadow brain2vec model using the known training method A. The adversary then extracts the same features accessible from M_{target} for each individual in U_{adv} using each shadow model. For each shadow model, there are $|U_{adv}| - n$ individuals that were not members of the shadow model's original pretraining data. A membership inference training dataset D_{sm} is created by extracting outputs for each participant in U_{adv} for each shadow model. The outputs are labeled a member if the individual was originally a member of the shadow model's training dataset, otherwise it is labeled a non-member. The adversary can then train a membership inference classification model C_{MI} using D_{sm}.

Our membership inference experiment is illustrated in Figure 47 and assumes that M_{target} is trained using only 2 participants for pretraining (i.e., 2-participant models). This is a necessary restriction of dataset size due to the need to have pretraining holdout participants to serve as non-members for both shadow modeling and evaluation of the attack using simulated target models. Therefore, we simulate an attacker with access to U_{adv} containing $|U_{adv}| = 4$ participants, allowing the attacker to build 6 shadow models from $\binom{4}{2}$ participant combinations. The participants in U_{adv} might represent additional public datasets or even individuals collaborating with the adversary to attack the target models, including the adversary themselves. The remaining 3 participants makeup U_{target} and are used to evaluate the adversary's C_{MI} with 3 target models from $\binom{3}{2}$ participant combinations. The participants and developers contributing

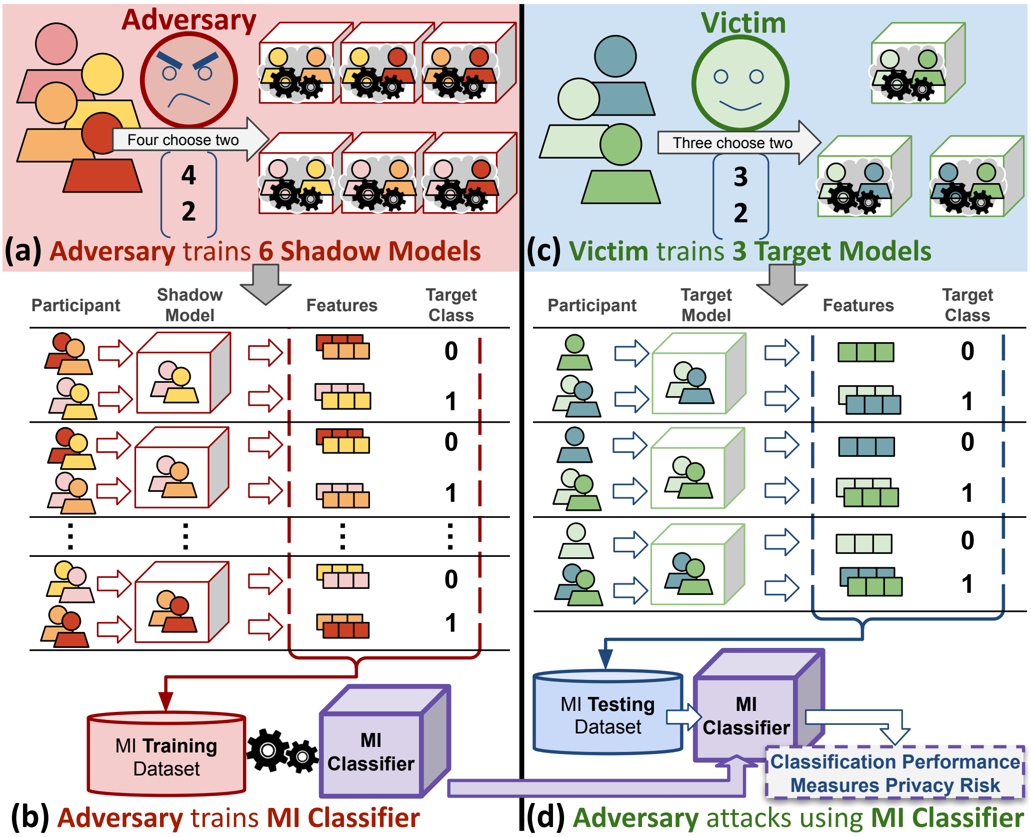


Fig. 47. **Method to Measure Privacy Risk of a Shadow Modeling Membership Inference (MI) Attack**: **(a)** From seven participants, four participants represent a cohort available to the adversary to produce shadow modeling datasets. Six unique shadow models from $\binom{4}{2}$ participant pairs are pretrained using the same method as the target model. **(b)** The adversary constructs a training dataset and trains a membership inference classifier by passing both members and non-members through each of the trained shadow models. **(c)** The remaining three participants represent contributors to the dataset used by victim, the person or people developing models that are targeted by the adversary. These participants desire that their contribution to a target model remain anonymous. These three participants are used to train three different private models from each of $\binom{3}{2}$ pairs, which will be used to evaluate the adversary's simulated attacks. **(d)** The resulting three target models are assumed to be "released" publicly and represents the adversary's end goal in our experiments.

to the target models are the hypothetical *victims* of the attack in Figure 47's illustration. The performance of the attacker's C_{MI} when predicting membership and non-membership from the three target models represents the membership inference risk under the outlined conditions.

For each pretraining configuration to be evaluated, $35 = \binom{7}{3}$ unique experiments are performed with $21 = \binom{7}{2}$ brain2vec models. To reduce computational burden of these experiments, we select the shadow modeling configurations based on the results of the re-identification experiment results rather than grid-search pretraining configurations. We believe this to be a sensible approach, since the individual-level correlations allowing re-identification may translate well to membership inference. Note that our re-identification task does not face this same computational burden - it reuses the 6-participant models produced in Section 5.3.2, and only $7 = \binom{7}{6}$ unique models must be pretrained for each configuration's re-identification experiment.

We explore multiple variations of inputs for the attacker's C_{MI} model. Given a frozen pretrained brain2vec model and a sensor sample, we begin by first producing the outputs Z_t from the brain2vec instance's feature encoder. We pass these features Z_t to the quantization module to produce the likelihood distribution over the possible codewords for each step (FQP). Next, we slide a mask of length 1 over the feature encoders output, producing the context network's prediction for each time step. For each of the context network's prediction, we capture the codeword likelihood distribution (CQP) or the contrastive loss L_c for each prediction (CL). Importantly, FQP and CQP are vectors of likelihoods equal in size to $|FQP| = |CQP| = T \times G \times W$. In order to reduce feature dimensions for the attacker model, we measure Shannon's entropy across codewords W for each codebooks G and time steps T, reducing dimensions to $|FQP| = |CQP| = T \times G$. We refer to these feature vectors as FQP-E and CQP-E. A summary reference of these outputs is provided below:

- Entropy of Feature Extractor's Quantization Probabilities (**FQP-E**): The entropy of the likelihood distribution of the codewords from quantizing the feature encoder's output.
- Context Encoders Quantization Probabilities (**CQP-E**): The entropy of the likelihood distribution of the codewords from quantizing the context networks masked predictions.
- Contrastive Loss (**CL**): The loss L_c for each of the context network's predictions. We take L_c as implemented for minimizing the similarity with distractors using binary cross-entropy during pretraining. To reduce skew in the distribution, we apply a simple negative-log transform f(x)=$-log(x + \epsilon)$, where $\epsilon = 1e^{-6}$.

Similar to the re-identification privacy assessment, the membership inference attack model (i.e., C_{MI}) must be applied to multiple individuals and their sensor configurations. We use the same approach for membership inference that we described for re-identification: we sample N_s sensors, without replacement, from each participant when a building a batch of data. Therefore, each C_{MI} model is trained to infer membership from a fixed set of sensors. We vary the number of sensors in our experiments to investigate if the risk of membership inference changes with the number of sensors producing features for the attacker.

Like the C_{RI} model, the C_{MI} model is meant to be a parsimonious baseline attempt at membership inference. The C_{MI} model is a fully connected feed-forward neural network with 2 hidden layers having 128 units and a final layer with 2 output units. An LReLU activation with negative slope of 0.01 is used in the hidden layers, followed by a softmax output in the output layer for classification. An element-wise dropout rate of .75 and batch normalization regularize the C_{MI} model.

5.3.4 Results

In this Section we report the results of the experiment workflow outlined in the prior sections and illustrated in Figure 44. We begin in Section 5.3.4.1 where we compare the average

Table 16. Average total loss across CV batches for 1-Participant pretraining

	Positional Encoding	Spatio-temporal from RAS		Spatial from RAS	
	Mask Length	1	2	1	2
Output Dimensions (Embed. x Time)	Codebook Size				
16x3	20	**579.60**	2728.97	789.45	**2138.34**
	40	976.46	**2184.61**	987.95	2208.49
	80	837.76	2262.12	**775.79**	2338.22
32x6	20	1483.70	1834.73	**1372.10**	**1720.91**
	40	**1409.19**	1884.00	1731.01	1798.62
	80	1412.54	**1631.00**	1446.82	1816.23

Table 17. Average CV loss for 1-Participant pretraining across encoder depth

			N. Encoder Layers	4	6	8
Output Dimensions (Embed. x Time)	Codebook Size	Mask Length	Positional Encoding			
16x3	20	1	Spatio-temporal from RAS	810.15	**579.60**	1043.35
		2	Spatial from RAS	2319.61	**2138.34**	6136.46
	40	2	Spatio-temporal from RAS	2292.03	**2184.61**	2419.35
	80	1	Spatial from RAS	**662.53**	775.79	705.93
32x6	20	1	Spatial from RAS	1621.10	**1372.10**	1510.18
		2	Spatial from RAS	2420.74	1720.91	**1645.86**
	40	1	Spatio-temporal from RAS	1489.41	**1409.19**	1457.75
	80	2	Spatio-temporal from RAS	1758.27	**1631.00**	1666.44

batch loss for the cross-validation partition of each 1-participant model after its completed the pretraining optimization procedure. We visualize the pretraining loss terms over the training epochs for various groupings of interest in order to characterize how the loss terms correlate with the hyperparamters. From these results, we select eight configurations to perform additional experiments for which we vary the encoder depth in additional experiments. We select the top performing configurations and pretrain 6-participant models using the same configurations. We then report their fine-tuning performance on the original speech tasks introduced in 5.2.1 using the selected configurations. Next, in Section 5.3.4.2, we report the 7-class accuracy for our re-identification experiment using the pretrained 6-participant models. Finally, in Section 5.3.4.3, we select a single configuration from the re-identification results to use in pretraining 2-participant models for our shadow modeling membership inference experiment. We report the balanced accuracy as our measure of membership inference risk.

5.3.4.1 Pretraining and Speech-related Task Fine-Tuning

The pretraining cross-validation loss L for the 1-participant grid search of hyperparameters is provided in Table 16. The average is taken across mean from each batch of the cross-validation data. Each cell represents the average total loss across all seven 1-participant models for that particular model configuration. Notably, configurations with a mask length of 1 always achieve lower loss scores than configurations with mask length of 2. Larger $32x6$ embeddings out-perform smaller $16x3$ embeddings when using a mask length of 2. Overall, the models with the lowest pretraining losses in our experiments used the smallest codebook, mask length, and feature extractor. The results bolded in Table 16 are the codebook sizes that achieve the lowest loss across the other hyperparameters.

For each of the bold configurations in Table 16, we optimize additional 1-participant pretrained models across context encoder depths of 4 and 8. Training losses for the encoder depth experiments are illustrated in Figures 51 and 52, and the average loss across the validation data is present in Table 17. For each selected configuration tested, we select the best performing encoder depth, shown as the bolded configuration in Table 17. These selected model configurations are reused in later experiments when exploring the impact of hyperparamter configurations.

The fine-tuning results for the selected model configurations, when pretrained with either 1 pretraining participant or 6 pretraining participants, are illustrated in Figures 54 and 55. In both figures, we group the fine-tuning participant on the "transfer type" - whether or not the participant was a member of the pretraining cohort. In Figure 54, we show results for configurations with a mask length of 1. In Figure 55, we show the same results for the selected configurations with a mask length of 2.

Figure 56 visualizes the accuracy of the 1-participant models across combinations of pretraining and fine-tuning participants. For brevity, only two well-performing models are visualized in this manner - one model selected from configurations with mask length of 1 and another configuration selected from results with mask length of 2 (full descriptions provided in the caption). The heatmaps appear more correlated for a specific fine-tuning participant, suggesting that the fine-tuning participant is the primary driver of fine-tuning performance.

The eight configurations selected from the 1-participant pretraining results were also pretrained with 6-participants. The pretraining losses for these models are illustrated in Figure 53. The fine-tuning results for these same models are shown alongside their 1-participant counterparts in Figures 54 and 55.

Table 18. Average balanced accuracy of Re-identification attacks on holdout

					Avg. Accuracy (StD.)
Output Dimensions (Embed. x Time)	Codebook Size	Mask Length	Positional Encoding	Num. Encoder Layers	
16x3	20	1	Spatio-temporal from RAS	6	0.258 (0.043)
		2	Spatial from RAS	6	0.251 (0.042)
	40	2	Spatio-temporal from RAS	6	0.263 (0.045)
	80	1	Spatial from RAS	4	0.262 (0.046)
32x6	20	1	Spatial from RAS	6	0.314 (0.050)
		2	Spatial from RAS	8	0.323 (0.054)
	40	1	Spatio-temporal from RAS	6	0.322 (0.050)
	80	2	Spatio-temporal from RAS	6	0.454 (0.167)

5.3.4.2 Participant Re-Identification

We take the 6-participant model configurations selected in the previous section and attempt to re-identify all participants, including the participant left out of each model's pretraining. The average configuration-specific re-identification accuracy on holdout samples, calculated across number of sensors N_s, can be found in Table 18. The aggregate re-identification attack accuracy for each N_s is illustrated in Figure 57. For visual clarity in Figure 57 we do not separate results by positional embedding method since we note low variance across this configuration. Instead, each facet in the Figure 57 represents a group of similarly configured brain2vec models, with color shading to indicate the configuration is a "Smaller" or "Larger" codebook. This is necessary due to the model selection process - only the better performing configurations were selected.

In Figure 58 we compare the re-identification performance between partiicpants who were members of the pretraining cohort and non-members. For this analysis, we used the class-level F1 score since we are separating the performance across classes. In other words, we are comparing the performance for re-identifying each participant individually, which is highly imbalanced (i.e., a positive class making up approximately $\frac{1}{7}$ of the samples), so we rely on a common measure suitable for this imbalance. Otherwise, layout and interpretation of Figure 58 is conceptually similar to Figure 57.

5.3.4.3 Membership Inference

We take the model configuration that achieves the best re-identification results in Section 5.3.4.2 for use as our target model configuration in our membership inference experiment. Specifically, we train 21 models on two pairs of the seven participants using a mask length of two, $W = 80$, six encoder layers, a spatio-temporal positional embedding from RAS, and 32x6 feature output dimension. We then used these 2-participant models to perform the membership inference experiment as previously outlined in Section 5.3.3.

The balanced accuracy for our membership inference experiments are illustrated in Figure 59. We compare the train and test partition performance to illustrate both over-fitting and the potential for the attack to generalize to its target model. We also provide a table of the same results in Table 19 We use balanced accuracy, defined as the average of recall for the two classes, due to the class imbalance in the holdout partition. The simple definition of balanced accuracy also makes it more suitable for measuring risk in an interpretable manner. Using balanced accuracy for our metric, even though the classes are imbalanced, makes it sensible for us to use a

Table 19. Average membership inference balanced accuracy on holdout

Features	CL	CQP-E	FQP-E
Number of Sensors			
1	0.510 (0.026)	0.500 (0.003)	0.499 (0.007)
2	0.524 (0.045)	0.501 (0.004)	0.503 (0.015)
4	0.519 (0.040)	0.501 (0.005)	0.504 (0.019)
8	0.533 (0.043)	0.499 (0.012)	0.502 (0.017)
16	0.537 (0.046)	0.502 (0.015)	0.514 (0.030)
32	**0.546 (0.059)**	0.501 (0.012)	0.511 (0.029)
64	0.553 (0.062)	0.505 (0.014)	0.512 (0.036)

50% accuracy as our baseline performance. Models must perform significantly better than 50% balanced accuracy in order to confirm a risk of membership inference. However, recent work in [126] argue that balanced accuracy should not be used for these evaluations, and that instead an analysis of the model's true-positive rate at thresholds producing low false-positive rates should be used instead. We leave this updated analysis technique to future efforts.

5.3.5 Related Work

This section's contributions are at the intersection of SSL, BCI, privacy, and security. We first refer to progress in vision and audio SSL to more broadly consider ways to evaluate pretraining methodologies. We then outline advances in BCI and why SSL methods such as brain2vec are unique in their capabilities and potential privacy challenges. Finally, we conclude with a review of literature related to privacy for machine learning models with focus on human physiology based systems.

Self-Supervised Learning: Recent SSL approaches use the implicit structure present in data to optimize a model during a "pretraining" phase. Importantly, SSL pretraining does not require a labeled dataset. The resulting pretrained model can later be applied to other "downstream" end tasks. These end tasks are typically supervised, often with much less data than was available for pretraining. Our work is primarily interested in the evaluation of SSL methods, rather than development of new SSL methodologies.

The authors of [257] present Visual Task Adaption Benchmark, which outlines and applies a methodology to compare visual SSL algorithms. Their focus is on diverse tasks and data sources, which cannot be easily replicated with our limited BCI data, but they do establish several relevant protocols. First, datasets for supervised task evaluation are expensive to collect and label, so it is permissible for evaluation of models to reuse the same tasks. Second, to avoid meta-overfitting, the pretraining algorithms should avoid using data that are a part of the evaluation tasks. Finally, a unified implementation should be used for the evaluation tasks - there should be no prior knowledge about the evaluation task implied by the architecture or parameter search. Their evaluation of pretraining in the visual domain demonstrates that pretrained models out-perform models trained from scratch for most evaluation tasks. Furthermore, they show that fine-tuning the entire model generally performs better than the more simple evaluation of appending a linear classification layer for training the evaluation task.

In [258] the authors compared thirty recent image-based SSL methods and how they perform when applied to downstream end tasks. They show that *structural* downstream tasks tend to benefit more from SSL than *semantic* tasks. In their experiments, structural tasks involved learning about the structures in the image, such as estimating depth or detecting walkable surfaces. Semantic tasks describes those involving labels of content in the image, such as image classification. The authors also show that starting with a self-supervised model, rather than another supervised task's encoder, tends to perform better as a feature extraction method. They also show that performance on downstream tasks is not always positively correlated with self-supervised training performance and that models pretrained from similar domains to their downstream tasks tend to lead to the best performance. They are also unable to to demonstrate the need for a well-balanced dataset when pretraining.

Brain-Computer Interfaces: Efforts to record and decode neural signals with BCI have have been progressing for decades [259]. In general, BCI solutions aim to improve medical treatment, enable or enrich human-computer interaction, and advance our understanding of neurological processes [260]. Distribution discrepancies force many BCI solutions to rely on per-patient models: approaches that, when deployed, are optimized from scratch for the individual. In these scenarios, any threat to an individual's privacy is localized to the new user and the management of their data. In contrast, deploying a model that has been pretrained on other individuals, threatens the privacy of all pretraining individuals through the possibility of information leakage. In order to study information leakage from pretraining BCI system, this work uses the *brain2vec* model presented in [42] since it's demonstrated ability to capture and utilize information from multiple participants' unlabeled neural data. The brain2vec model has been assessed for neural decoding of speech-related tasks, which is closely related to other prior work in BCI [195, 189, 196, 197, 198, 199, 200, 192, 261, 202, 203, 204, 205].

Physiological security: Adversarial attacks on physiological computing are surveyed in [262], illustrating the growing interest in attacking the integrity of such systems with modified data that leads to incorrect behavior. The authors note the popularity of EEG and that methods for attacking transfer learning methodologies have been limited. Still, the work reviewed is focused on attacking systems in use, but no authors have investigated the potential privacy risks in transfer learning from physiological data.

Advancements in BCI and other physiological systems have further motivated research into *neurosecurity* - the application of computer security and privacy concepts to neural devices and their communications [263]. The recent survey by [264] outlines the progress in neurosecurity, describing it as a nascent field with four key areas: integrity, availability, safety, and confidentiality. Attacks on BCI integrity are no different than attacks on any other information system. Within the application of BCI, integrity attacks attempt to modify or destroy neural-related data, and can target data at point of storage, transmission, or receipt. Reducing availability reduces the scope of the authorized users access, degrading the BCI's utility and potentially causing other harms [265, 15]. Efforts to degrade the safety of BCI systems aim to harm users, including physiological or even psychiatric harms [266]. Confidentiality of BCI systems is increasingly important as they become more successful at decoding thoughts, intentions, and aspects of an individuals character [265]. Confidentiality - how participation in a BCI systems construction may enable future violation of a users privacy - is a primary focus of the work in this section.

Privacy and Information Leakage in ML: In order to assess the potential for privacy violations in BCI, we apply approaches to assess information leakage in ML approaches first applied in other domains. The work in [25] introduced the adversarial framework of shadow models for membership inference attacks. Using only the predicted class likelihoods, their efforts demonstrated highly accurate membership inference attacks across both tabular and image modalities. The authors in [267] investigate the information discernible from the updates to language models, enabling an attacker to understand the differences in training dataset used to train two language models. Recent work in [268] proposed unified general measure for quantifying information leakage through membership inference attacks. They introduce *Nirvana*, a methodology which minimizes generalization gaps to help prevent membership inference white-box attacks. Both white-box and black-box attacks are considered for graph neural networks in social network setting by the authors in [269]. Privacy methods have also been applied to the HAR domain. The authors of [270] applied shadow modeling techniques to HAR to estimate information leakage from inertial sensors in a mobile device. In [271], researchers propose a framework for video-based activity recognition. Their approach uses an adversarial modeling process to anonymize video frames by modifying their content to remove identifying information yet still contain the relevant activity.

5.3.6 Discussion

We produce hundreds of experiments in order to investigate brain2vec's privacy and performance, which we discuss in more detail in this section. First, our pretraining grid-search yields insights into how the pretraining losses are impacted by various changes in architecture. Next, our re-identification experiments illustrate brain2vec's ability to identify individuals, even if only given random subsets of their sensors. We are also able to show that certain model configurations result in better re-identification performance. Finally, while our membership inference experiment shows poor generalization for most configurations, we are able to illustrate that certain features have increased risks when access to the number of sensors increases.

In our **pretraining grid search experiments**, the largest impact on the pretraining loss terms result from varying the dimensionality of the output by reducing the size of the feature extractor module at the input of brain2vec. Changes to these dimensions impact all the operations that follow, including quantization and inference of the masked time-step, so it is unsurprising that it has a large impact on optimization. These differences can be seen in Figures 48, 49, and 50. In particular, Figure 50 is the most clear - our smaller output dimension (16x3) achieves a better contrastive loss than the larger output dimension (32x6) when a mask length of 1 is used, but performs worse with a mask length of 2. This is likely because the smaller dimension does not provide enough information in a single time-step in order to support the inference of the other two. Whereas when trained using the larger dimension, 4 time-steps are used by the context encoder (i.e., 6 steps with 2 masked) to predict a vector that is similar to the quantizer's output. Indeed, we can see that the smaller output dimension also poorly utilizes the codebook of the model, since the diversity loss generally performs better when the output dimension is larger. We find that smaller codebooks (i.e., smaller W) generally achieve lower diversity loss since dimensionality over which diversity is measured is reduced. However, we note that the smaller codebook does not appear to impact the contrastive and feature penalty terms in our experiments. The feature penalty increases with a 32x6 dimension output since there are more terms that make up the penalty, though the larger model appears to begin to reduce its feature penalty earlier and more rapidly than its 16x3 counterparts. Illustrated in Figure 53, we find similar results regarding the configuration of the feature extractor when pretraining 6-participant models. Due to the cost of pretraining these larger models, we only pretrain eight of the best performing configurations, which makes it difficult to compare model configurations due to the number of hyperparameters that are varying. However, as can be seen in Figure 53's two columns of training loss curves, the distinction between the performance of the two output dimension configurations continues even with increased data.

We find that the changes to the positional encoding have little impact on the pretraining losses for the smaller 16x3 model. However, for the larger model, our results suggest a lower diversity loss with a spatio-temporal embedding derived from the RAS coordinates. When only using RAS for the spatial component instead, the diversity loss appears to actually worsen. This relationship can be seen in several ways across the pretraining losses presented in Figures 48, 49, and 50. We hypothesize that this may be due to the difficulty of disentangling the positional component in order to produce features similar to the quantizer, which has no positional awareness. This assumes that decoupling the temporal component into its own parameters results in a more complex positional representation. However, if this were the case, we might expect improvement from a larger context network, yet as shown in Figures 51 and 52, we find very little impact to pretraining performance metrics when varying the number of transformer encoder layers in the context network.

As shown in Figures 54 and 55, we find inconsistent variance between the pretraining configuration and the performance of downstream classification tasks. This overall lack of correlation between pretraining performance and downstream fine-tuning performance in our experiments has also been illustrated in the visual domain for some models and tasks [258]. In our work, it appears that speech activity recognition performs better with the smaller output dimension, but larger models with larger mask lengths in Figure 55 do better at speech-related behavior recognition. The word classification task appears to be the noisiest, likely due to it's small sample size and increased number of output classes. It's also important to note that the sample sizes of the "Pretain Pt." and "Non-Pretrain Pt." in Figures 54 and 55 varies between the columns of pretraining cohort size. Specifically, 1-participant brain2vec models have a larger non-pretrain cohort ($N = 6$) and smaller pretrain cohort ($N = 1$) for evaluation. Of course, this is reversed for the 6-participant models, which have a smaller non-pretrain cohort ($N = 1$) and a larger pretrain cohort ($N = 6$). These differences in sample sizes should be better controlled in future evaluations, possibly regressing downstream classification performance onto hyperparameters using a linear model in order to better gauge the effect. We leave this task to future work.

Our pair-wise comparison of pretrain-to-fine-tune performance in Figure 56 helps illustrate the inconsistent performance. Figure 56 shows how fine-tuning performance can be primarily driven by the selected fine-tuning participant in the speech-related behavior and speech activity detection tasks (left and center column plots) - these tasks have a more clear tendency to have correlation across rows (pretraining participant) in comparison to the word classification task. From these sample visualizations, we further demonstrate that the relationship between the pretraining participant and the participant used for down-stream fine-tuning is highly variable and inconsistent across our three supervised tasks.

Our **participant re-identification experiments** clearly demonstrate the capability of brain2vec as a tool for detecting individuals using their brain recordings. Even with only a handful of *randomly selected* sensors, a simple model can be used to re-identify individuals well above the baseline class rate. Our results show the increased privacy risk of sharing more sensors with the attacker - the accuracy in each of Figure 57's and 58's subplots is directly correlated with the number of sensors used. Our results also show that higher dimensional features of 32x6 also increase the re-identification accuracy. Though we don't demonstrate consistent difference in re-identification success across mask-length, we observe high-performing outliers only in configurations with mask length of 2 and larger codebooks.

It's intuitive that this would be possible since no aspect of brain2vec's pretraining regimen is intended to penalize features that are correlated with the participant that provided the input sample. However, while there is also no aspect of brain2vec's pretraining that specifically encourages correlation with the participant, there are many aspects of the data that will correlate. For example, the sensor location, provided as RAS coordinates to the model, can vary widely across participants, as illustrated earlier in Figure 45. These discrepancies, as well as others like the spectral distribution across participants, are well known challenges in HAR problems. Since these distributions vary across participants, they can act as a "fingerprint", helping to uniquely identify the individual. In our case, the features extracted by brain2vec enabled 7-class accuracy of over 50% in some cases. Our results were achieved without substantial grid-search of attacker model hyperparameters - future work may easily improve on these results with larger hyperparameter grids or other improved designs.

The **membership inference experiment** does not generalize well to the simulated target model, suggesting that membership inference with our approach to be difficult. As shown in 59, we find that the FQP-E and CQP-E features perform similarly, but FQP-E appears to over-fit less, illustrated by its lower train performance and slightly higher test performance. Still, both of these features are unable to achieve performance on the test set that is significantly above a 50% balanced accuracy. In other words, we are unable to demonstrate any non-negligible risk of membership inference using FQP-E and CQP-E as attack vectors.

In contrast, we find that the CL features appear to pose some risk for membership inference. Figure 59 also shows that the attacker's C_{MI} model performs equally well on the train partition when using CL as it does when using FQP-E and CQP-E features. However, when using CL, the attack has a clearly increasing performance in the test partition as more sensors are made accessible. While the resulting accuracy in the test partition are still very low for CL, only reaching about 55%, the performance appears to be significantly above the baseline and the performance of FQP-E and CQP-E.

Each configuration performs better during training as they gain access to more sensors, but only the CL features demonstrate any ability to beat our baseline of 50% balanced accuracy. The attack models are over-fitting to the shadow modeling dataset, suggesting that the features are information rich for the goal of membership inference, but they don't generalize well to unseen target models. In general terms, the membership inference experiment tests the change in response when a self-supervised model is applied to data outside the pretraining distribution. In order to construct a reliable attack, the classifier C_{MI} is tasked with detecting in-distribution data, and therefore must also recognize out-of-distribution data. As shown by the high performance in the training partition, it is clearly possible to perform this task if given access to instances of the target model. But even with multiple examples of out-of-distribution responses in the shadow modeling dataset (i.e., the train partition), the attacker struggles to discover a reliable general-purpose mapping for their attack (i.e., the test partition). We hypothesize that the self-supervised pretraining and its randomly initialized target structure (i.e., the codebook) may be guarding against our membership inference attack. It may be that the learned representation, driven by the input data and quantization strategy, are likely to vary considerably from model instance to model instance, even with identical data. This is because the pretraining procedure does not enforce a specific structure, nor does it encourage interpretable features in anyway.

While our membership inference results don't illustrate a high-risk for membership inference attacks against brain2vec, the performance of CL features does motivate further research. With larger datasets and more methodical attack model development, the feasibility of such an attack may increase. It is the responsibility of researchers to continue to evaluate how well their models withstand these attacks, and to communicate those results to contributing individuals.

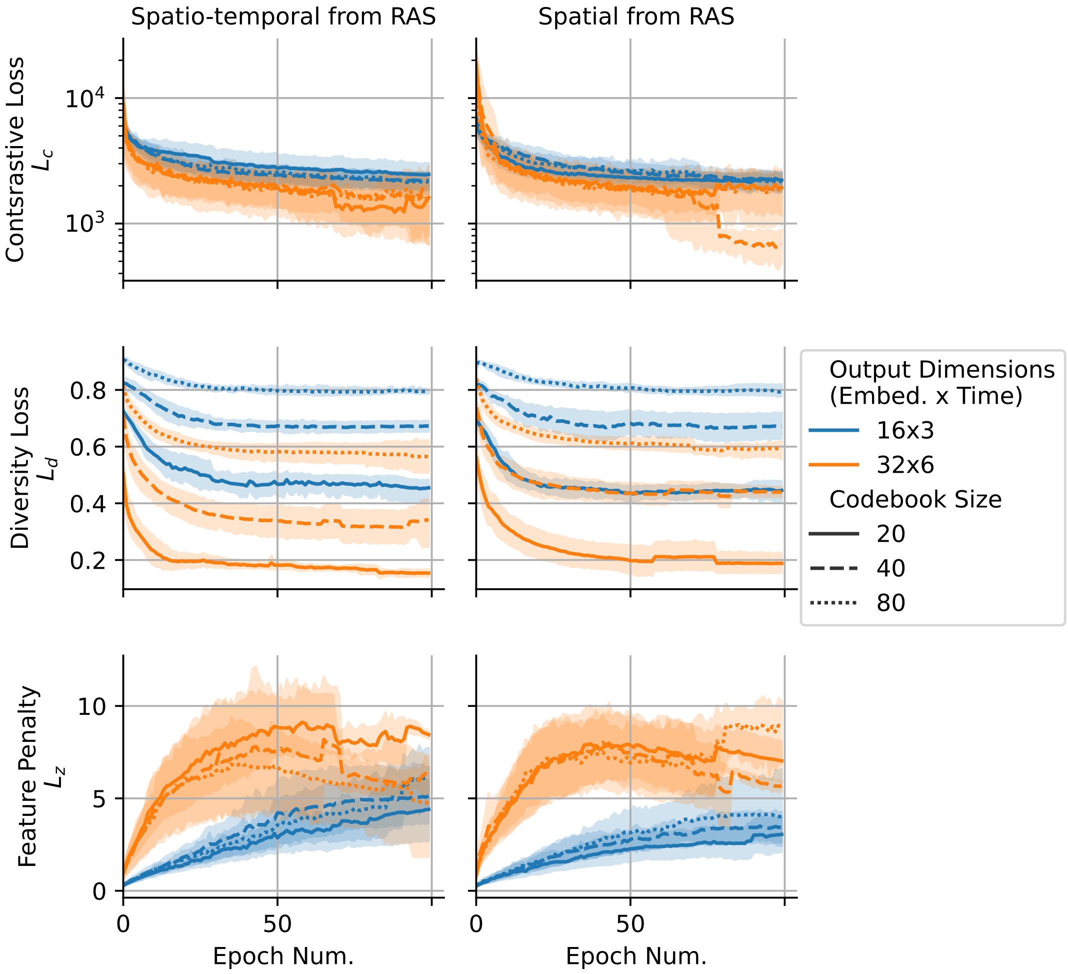


Fig. 48. Pretraining results on individual participants using mask of length 2. Each model configuration is pretrained 7 times, once for using each participant's data. Each row is a different loss term over the training epochs. Contrastive loss, top row, tends to perform better with smaller quantization configurations and RAS-based positional embedding.

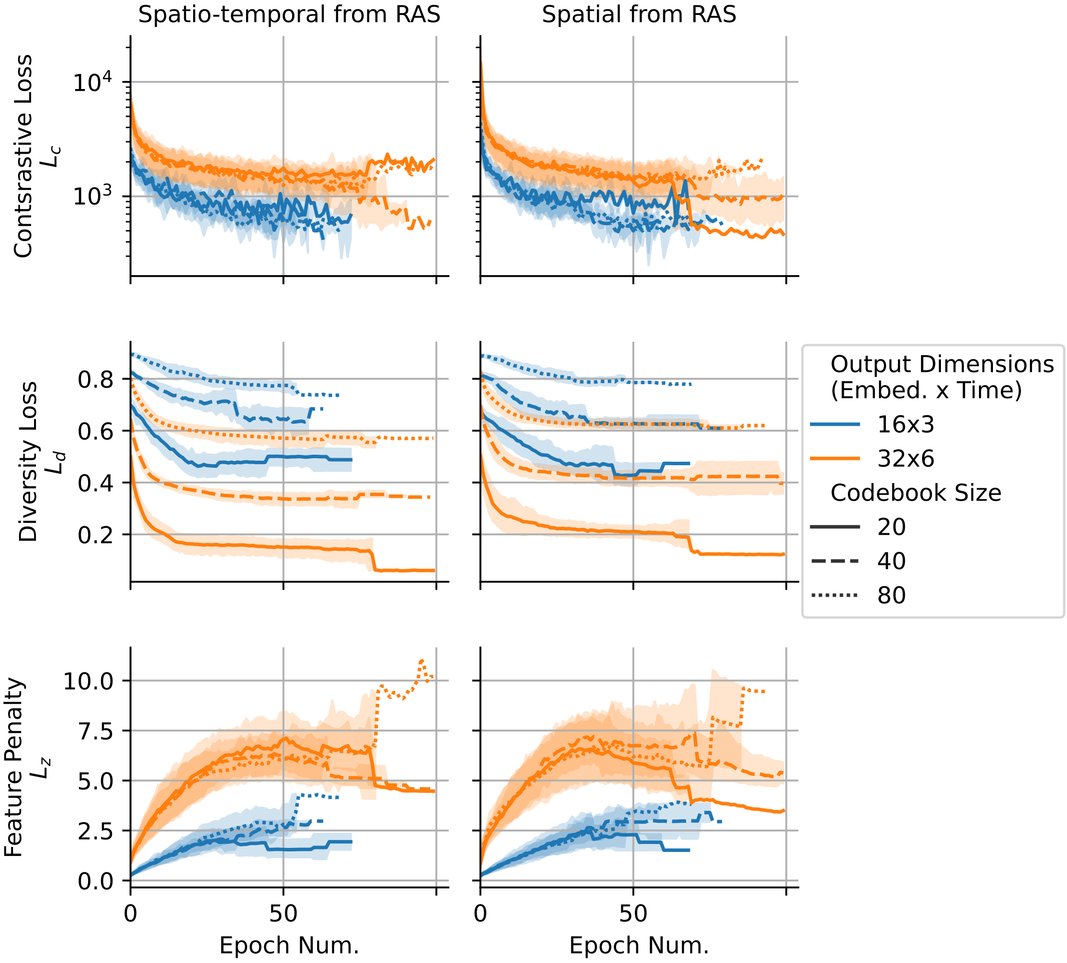


Fig. 49. Pretraining results on individual participants using mask of length 1. Each model configuration is pretrained 7 times, once for using each participants data. Each row is a different loss term over the training epochs. Contrastive loss, top row, tends to perform better with smaller quantization configurations and RAS-based positional embedding.

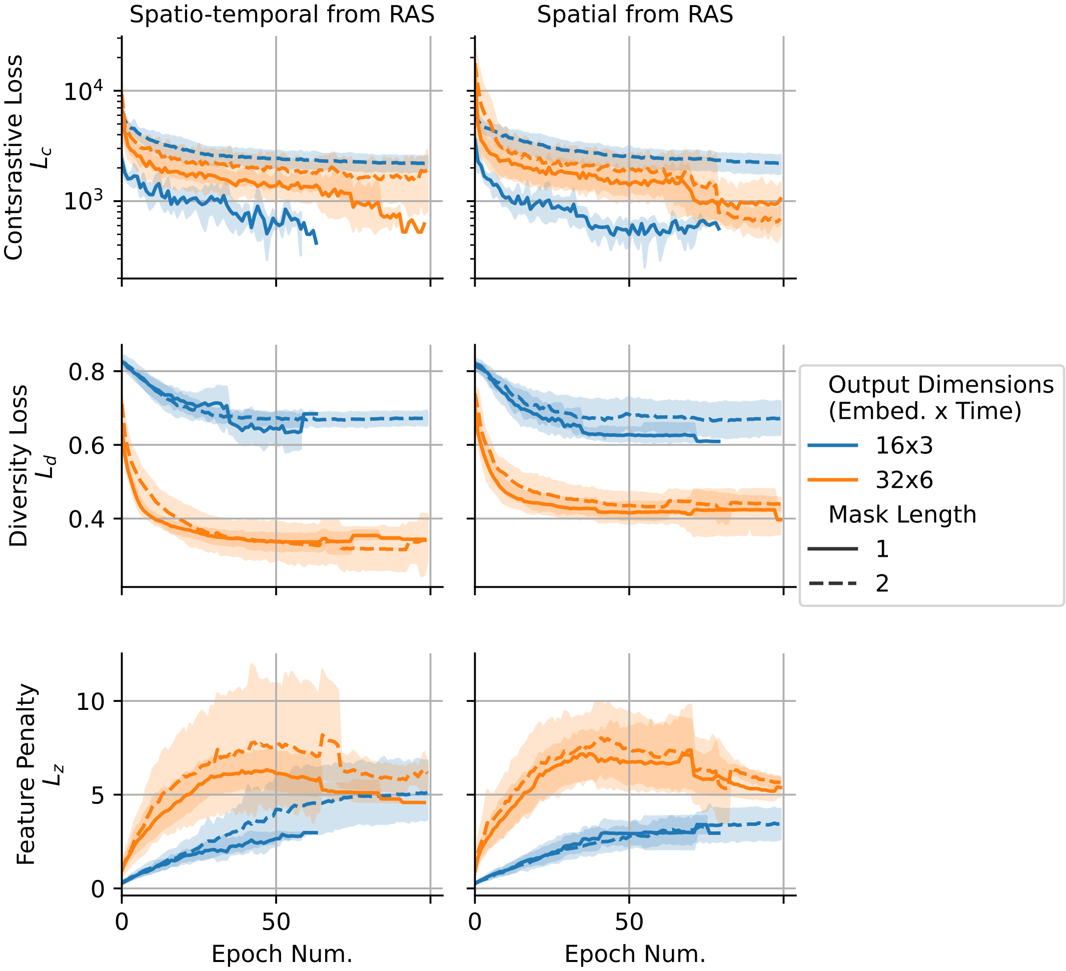


Fig. 50. Pretraining results on individual participants using quantization depth of 40. Each model configuration is pretrained 7 times, once for using each participants data. Each row is a different loss term over the training epochs.

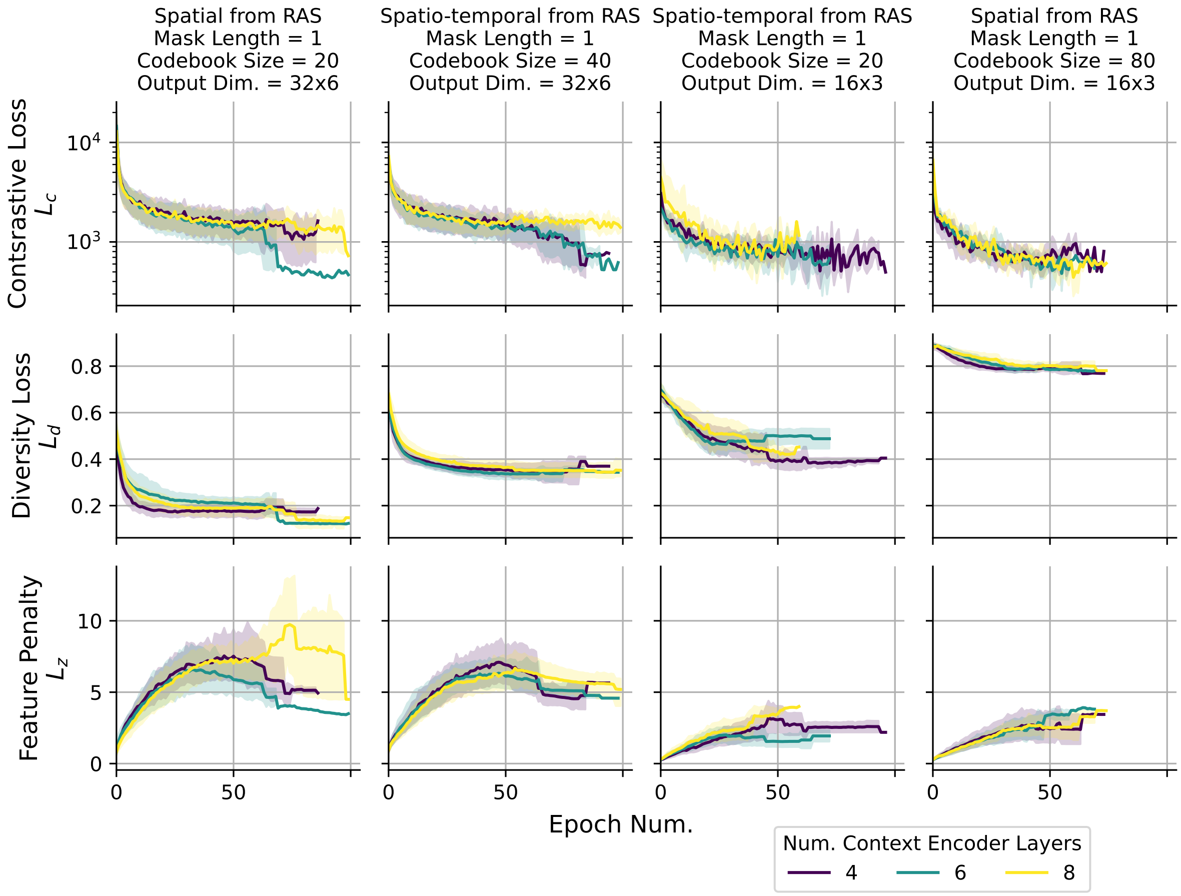


Fig. 51. The best performing 1-participant configurations with mask length of 1 optimized with context encoder depths of 4, 6, and 8 layers. We observe little difference between depth configurations.

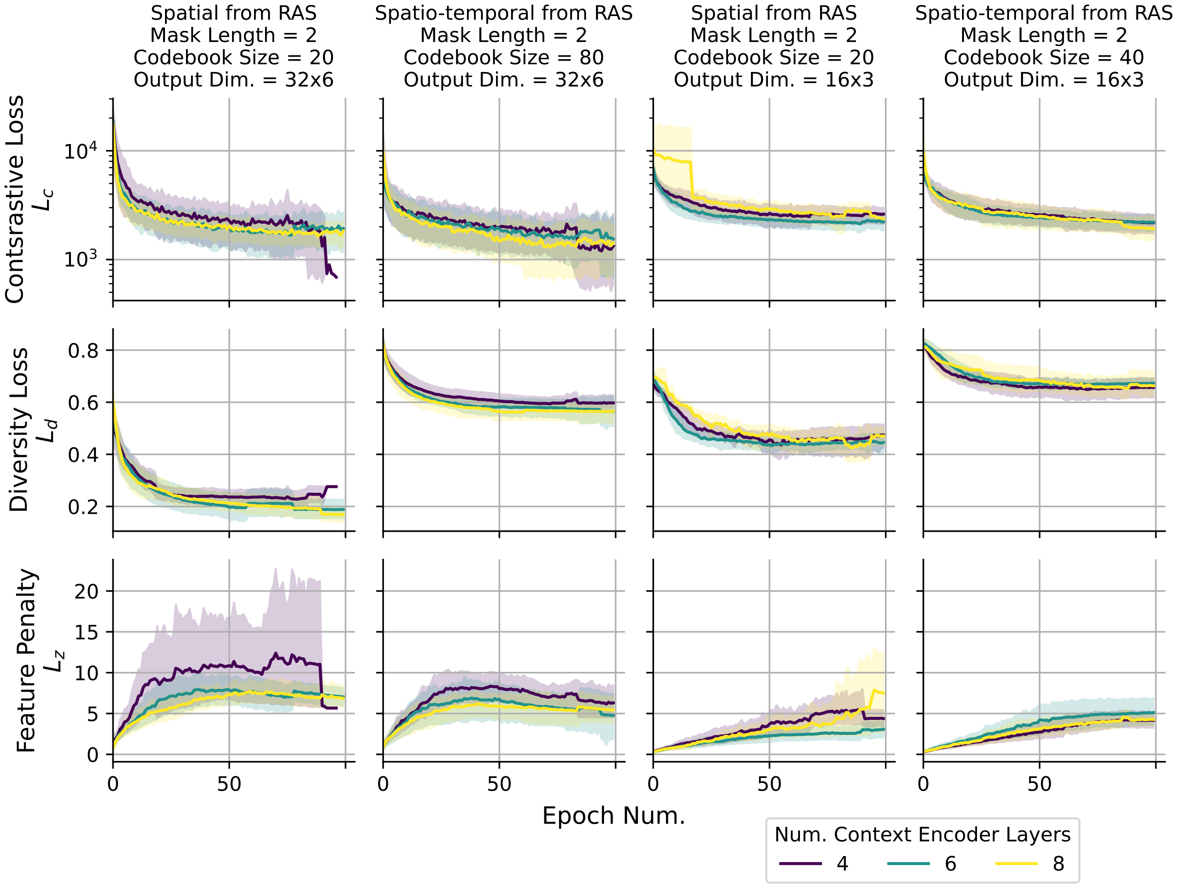


Fig. 52. The best performing 1-participant configurations with mask length of 2 optimized with context encoder depths of 4, 6, and 8 layers. We observe little difference between depth configurations.

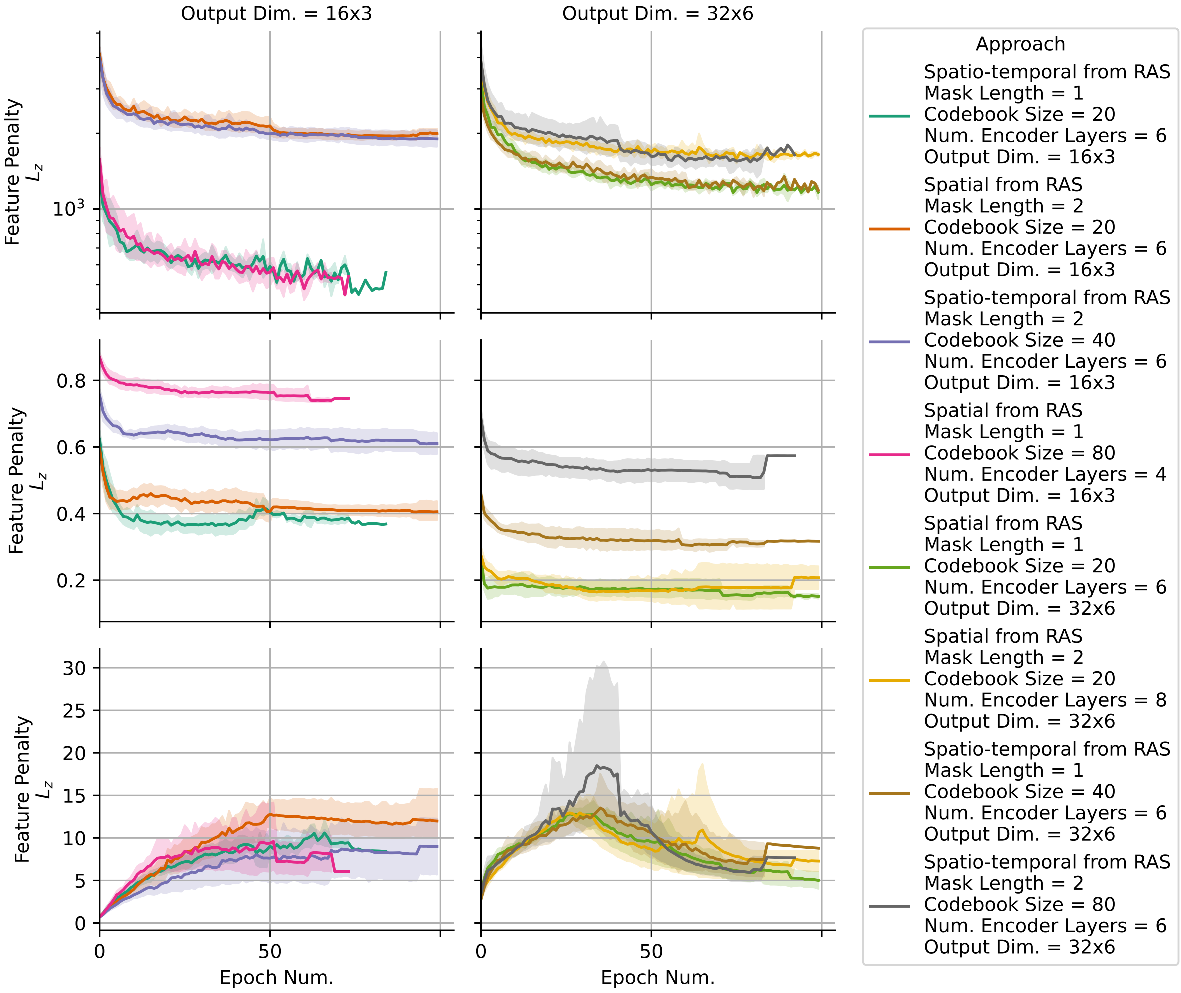


Fig. 53. Pretraining losses for 6-participant models configured using the best performing 1-participant model configurations. We highlight the variation in performance with respect to the change in the feature extraction's dimensions by separating this hyperparameter into columns.

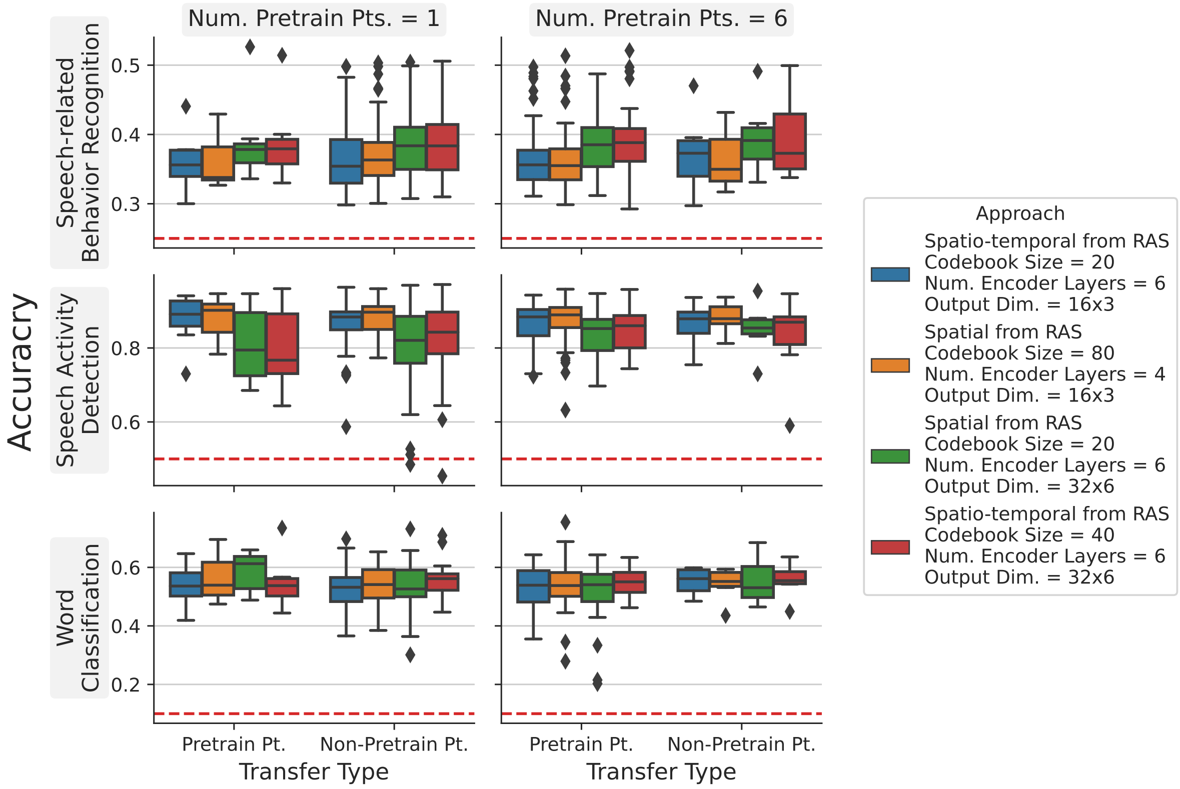


Fig. 54. Fine tuning results of the selected brain2vec models with mask length of 1. The "Pretraining Pt." are results from fine tuning participants that were used to pretrain the model, while "Non-Pretrain Pt." refers to results from fune-tuning participants that were new to the model during fine-tuning.

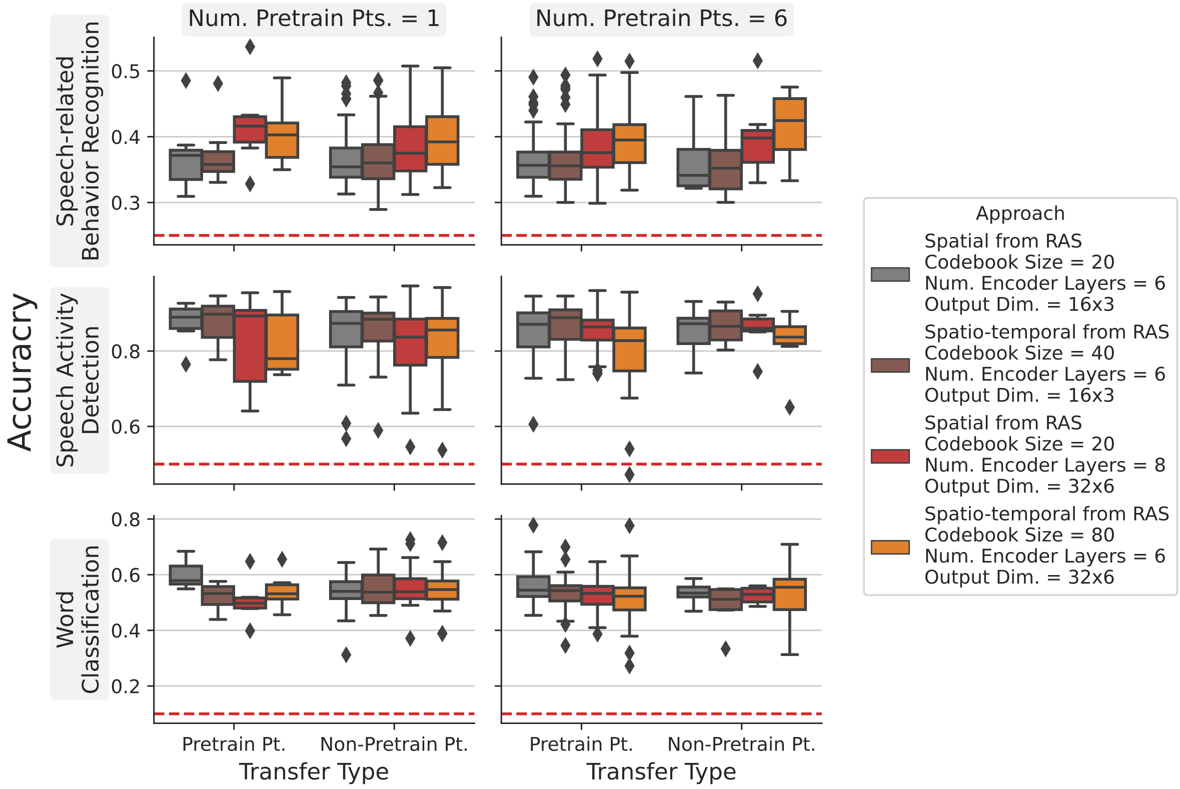
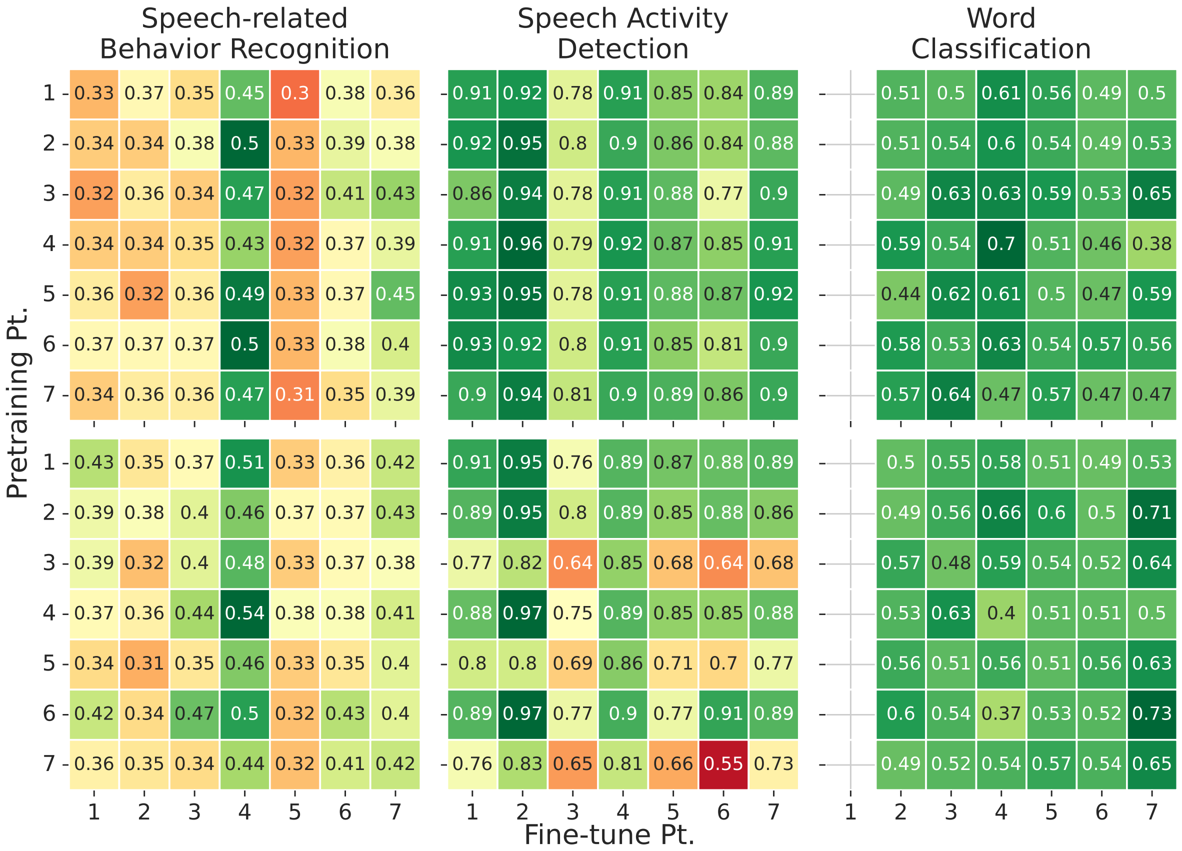


Fig. 55. Fine tuning results of the selected brain2vec models with mask length of 2. The "Pretraining Pt." are results from fine tuning participants that were used to pretrain the model, while "Non-Pretrain Pt." refers to results from fune-tuning participants that were new to the model during fine-tuning.



Pretraining Pt.	1	2	3	4	5	6	7
1	0.33	0.37	0.35	0.45	0.3	0.38	0.36
2	0.34	0.34	0.38	0.5	0.33	0.39	0.38
3	0.32	0.36	0.34	0.47	0.32	0.41	0.43
4	0.34	0.34	0.35	0.43	0.32	0.37	0.39
5	0.36	0.32	0.36	0.49	0.33	0.37	0.45
6	0.37	0.37	0.37	0.5	0.33	0.38	0.4
7	0.34	0.36	0.36	0.47	0.31	0.35	0.39

Pretraining Pt.	1	2	3	4	5	6	7
1	0.91	0.92	0.78	0.91	0.85	0.84	0.89
2	0.92	0.95	0.8	0.9	0.86	0.84	0.88
3	0.86	0.94	0.78	0.91	0.88	0.77	0.9
4	0.91	0.96	0.79	0.92	0.87	0.85	0.91
5	0.93	0.95	0.78	0.91	0.88	0.87	0.92
6	0.93	0.92	0.8	0.91	0.85	0.81	0.9
7	0.9	0.94	0.81	0.9	0.89	0.86	0.9

Pretraining Pt.	1	2	3	4	5	6	7
1		0.51	0.5	0.61	0.56	0.49	0.5
2		0.51	0.54	0.6	0.54	0.49	0.53
3		0.49	0.63	0.63	0.59	0.53	0.65
4		0.59	0.54	0.7	0.51	0.46	0.38
5		0.44	0.62	0.61	0.5	0.47	0.59
6		0.58	0.53	0.63	0.54	0.57	0.56
7		0.57	0.64	0.47	0.57	0.47	0.47

Pretraining Pt.	1	2	3	4	5	6	7
1	0.43	0.35	0.37	0.51	0.33	0.36	0.42
2	0.39	0.38	0.4	0.46	0.37	0.37	0.43
3	0.39	0.32	0.4	0.48	0.33	0.37	0.38
4	0.37	0.36	0.44	0.54	0.38	0.38	0.41
5	0.34	0.31	0.35	0.46	0.33	0.35	0.4
6	0.42	0.34	0.47	0.5	0.32	0.43	0.4
7	0.36	0.35	0.34	0.44	0.32	0.41	0.42

Pretraining Pt.	1	2	3	4	5	6	7
1	0.91	0.95	0.76	0.89	0.87	0.88	0.89
2	0.89	0.95	0.8	0.89	0.85	0.88	0.86
3	0.77	0.82	0.64	0.85	0.68	0.64	0.68
4	0.88	0.97	0.75	0.89	0.85	0.85	0.88
5	0.8	0.8	0.69	0.86	0.71	0.7	0.77
6	0.89	0.97	0.77	0.9	0.77	0.91	0.89
7	0.76	0.83	0.65	0.81	0.66	0.55	0.73

Pretraining Pt.	1	2	3	4	5	6	7
1		0.5	0.55	0.58	0.51	0.49	0.53
2		0.49	0.56	0.66	0.6	0.5	0.71
3		0.57	0.48	0.59	0.54	0.52	0.64
4		0.53	0.63	0.4	0.51	0.51	0.5
5		0.56	0.51	0.56	0.51	0.56	0.63
6		0.6	0.54	0.37	0.53	0.52	0.73
7		0.49	0.52	0.54	0.57	0.54	0.65

Fig. 56. Fine tuning results of two 1-participant pretrained models. In the top row: mask length of 1, a codebook size of 80, 4 encoder layers, and an embedding vector size of $16x3$. In the bottom row: mask length of 2, a codebook size of 20, 8 encoder layers, and an embedding vector size of $32x6$. Along the y-axis of each heatmap is the participant ID whose data was used to pretrain the model. The x-axis provides the participant ID that was used to fine tune the model supervised fashion. Values in each cell are the holdout accuracy and the shading is centered at 1.5x each task's target rate.

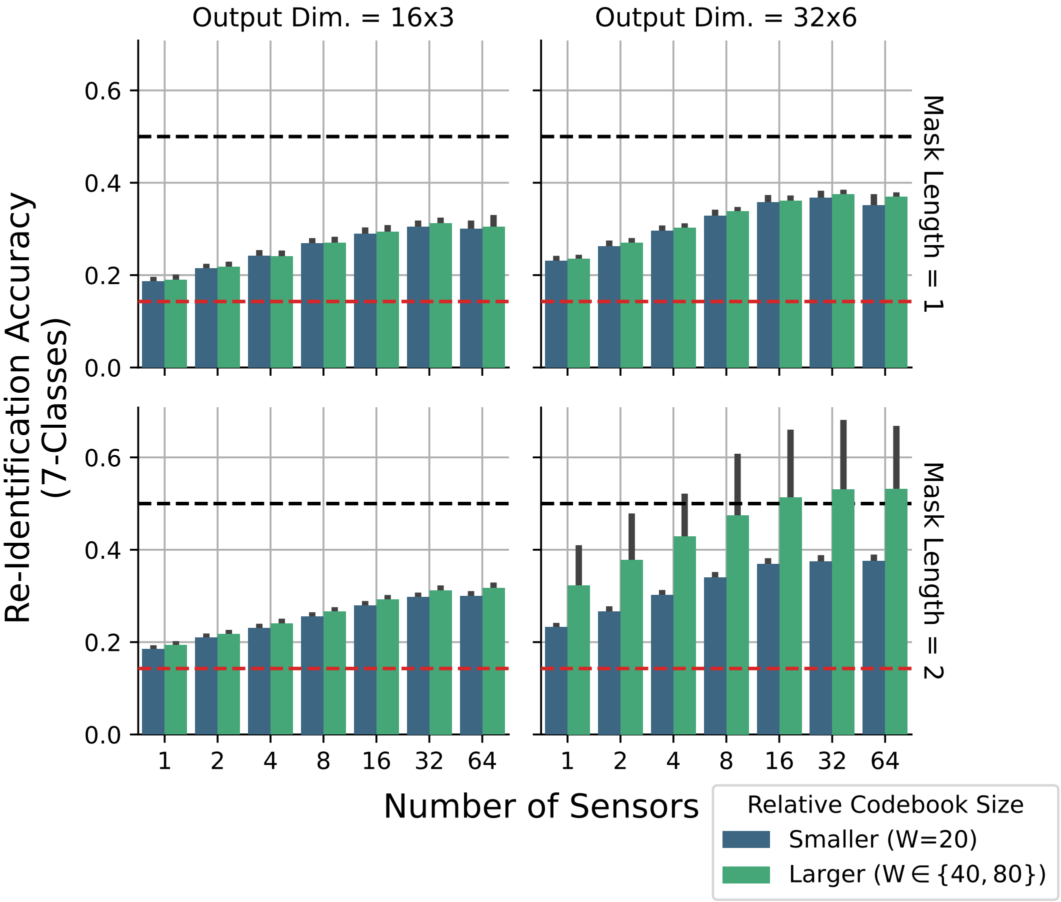


Fig. 57. Re-identification accuracy using 6-participant pretrained brain2vec models. In all cases, increasing the randomly sampled sensors (X-axis) improves test accuracy (Y-axis). However, the larger output dimension (i.e., two plots in the right column) achieves higher performance overall. Furthermore, high performing outliers only occur in configurations with mask length of 2 (i.e, lower right plot), achieving nearly twice the accuracy of comparable configurations. The variance in the bottom right is tied to three instances of pretrained brain2vec models with a codebook size of 80, each of which performs much better than other instances with identical configuration.

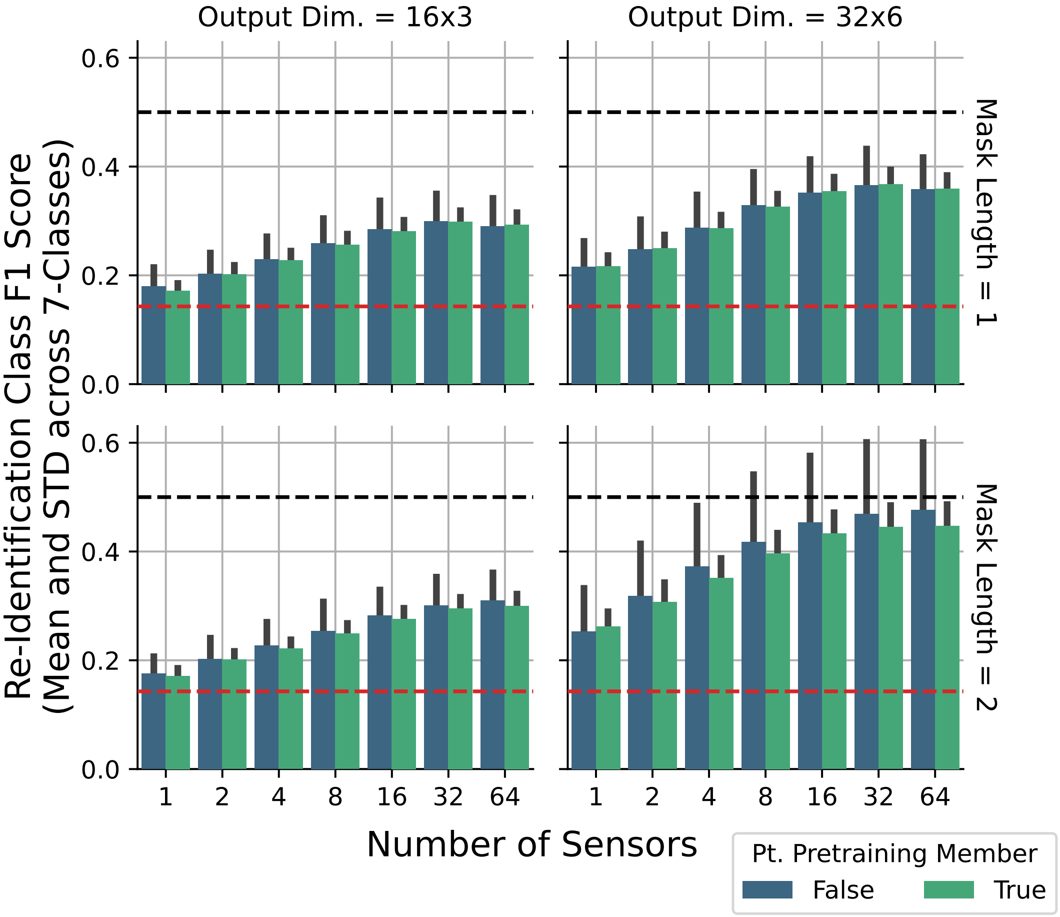


Fig. 58. Re-identification F1-score averaged across classes (i.e., participants), separated by whether data from the class was seen during pretraining. Models with different codebook sizes are aggregated together for the purpose of this visualization. Results show higher variance and slightly higher performance for participants not in the pretraining data, though differences don't appear significant in most cases. Results shown are in concordance with the 7-class accuracy shown in Figure 57 - primarily that the larger model with larger masks appears to improve re-identification performance.

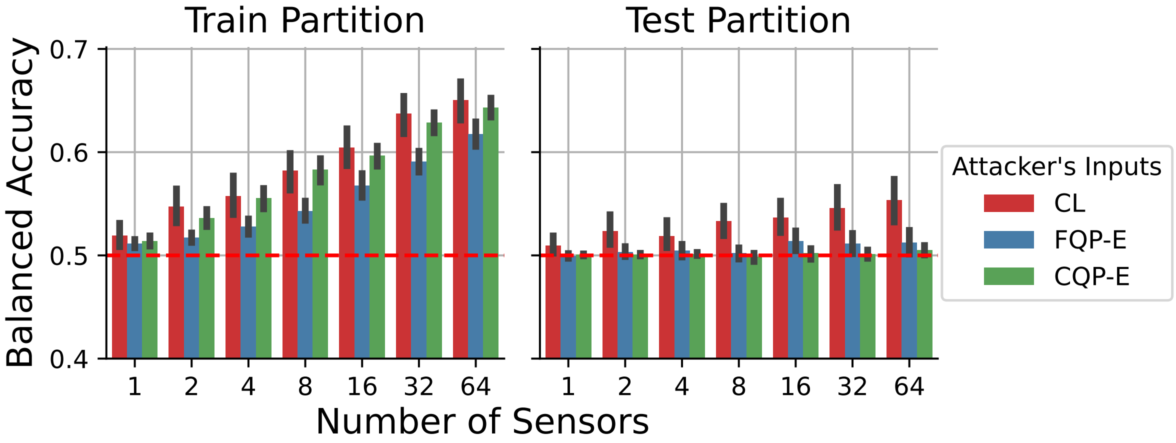


Fig. 59. Balanced accuracy from our membership inference experiment described in Section 5.3.4.3. We use balanced accuracy due to the class imbalance. The membership prediction performance on the "Train Partition" generally performs well, but the performance on the "Test Partition" remains at random chance for binary classification for FQP-E and CQP-E features. However, there appears to be a clear trend in the performance of the CL features, in which the performance increases to significant levels when given access to more sensors.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This dissertation was motivated by the lack of progress in HAR model designs that prioritize trust and adaptability when applied to bioelectric signals. While HAR solutions are important to enabling larger HCI and scientific research goals, ML-based approaches have often been borrowed from separate domains such as computer vision. However, naively transferring designs between domains results in overly complex solutions that ignore issues like usability, interpretability, and adaptability. Our contributions were designed to reduce complexity, increase interpretability, and enable models to transfer knowledge to new users and tasks. To improve trust, we use engineering informed designs to integrate interpretability into the model and greatly reduce complexity. To improve adaptability, we use transfer learning and self-supervised learning with person and task specific fine-tuning to transfer knowledge within the model. Below we enumerate each of our goals with a brief discussion of this dissertation's contribution.

Goal 1: Improve interpretability and reduce complexity by learning engineering-informed models

Our first goal in this dissertation was accomplished in Chapter 3 and 4. In Section 3.2, we illustrate an approach to interpretable preprocessing of bioelectric signals for use in ML algorithms detecting effective DBS treatment. With this work, we are the first to implement a classifier detecting optimal DBS using EEG in a post-operative setting. Critical for such novel work is that the method allows subject matter experts to validate that the model is using features congruent with their understanding of the problem. Without this capability, experts may be concerned that information leakage or spurious correlations are driving results, rather than generalizable features associated with the brain's underlying mechanisms. Therefore, our modeling pipeline is designed to select interpretable features from a collection extracted based on expert guidance. We find that our best performing models tend to use motor-related brain regions for classification, which aligns well with expectations since DBS was used to treat motor-degenerative disorders in our study. Our methods are evaluated on the challenging task of classifying the response in participants completely held out from model training, and our models consistently beat baseline class rates.

We continue with our primary contribution for interpretable modeling of bioelectric signals in Section 3.3 with SincIEEG, a model for speech detection from neural signals collected using IEEG sensors. Prior work for speech detection relies on expert guided features and preprocessing, similar to our own work in Section 3.2. However, we desired to move passed this paradigm and develop models capable of discovering these features directly from data, with minimal expert guidance. To that end, SincIEEG uses an engineering-informed input layer we developed called Multi-SincNet that borrows well known mathematical properties of periodic signals to learn interpretable input features. Trained on each individual participant separately, the model learns a filter bank of FIR bandpass filters. The resulting model therefore has a globally interpretable input layer integrated into the model architecture, which we use to help validate the model by comparing the interpretable features with previous research findings. We find that the features automatically discovered by our method align well with prior research that relied on grid-searches of extracted features. The use of bandpass filters also reduces the number of parameters needed compared to more traditional CNNs or RNNs applied in prior work, making our solution more usable in practice. Our method's performance meets prior work's classification performance, but without the need for expert-guided feature extraction and expensive preprocessing.

In Chapter 4, a hand-pose classifier using EMG data is implemented using a similar method to SincIEEG which we call SincEMG. Prior work for HAR on EMG has relied on expensive feature extraction or difficult to interpret deep learning models, but our SincEMG design allows us to visualize how learned features evolve from pretraining all the way to fine-tuning on new users. We implement our method on existing datasets that already had extensive prior work for our comparison. Our SincEMG solution beats prior work's classification performance, but with no preprocessing and an order of magnitude fewer parameters than previous methods, greatly improving usability. The improved interpretability and reduced complexity of our contributions supports this dissertations trustworthy objectives.

Goal 2: Improve inter-person adaptation using transfer learning across individuals

The second goal of this dissertation was achieved in Chapters 4 and 5. In Chapter 4, we implement transfer learning for HAR on EMG to pretrain SincEMG in a supervised fashion, then SincEMG is fine-tuned using the same classification task for the new individual user. Prior work used large models which we argue are more applicable in image processing domains. Our methods out-perform the state-of-the-art, in-part due to a novel application-aware regularization scheme that aggressively augments data during training to prevent over-fitting in both pretraining and fine-tuning stages.

In Chapter 5.2 we introduce brain2vec, a self-supervised learning method for speech-related HAR tasks using bioelectric signals recorded from sEEG sensors. Rather than rely on more traditional supervised learning, brain2vec is inspired by recent advances in natural language modeling and representation learning. Prior work, similar to our own in Section 3.3, often relies on patient-specific models, increasing the cost of developing practical HAR solutions since every user requires a model trained "from scratch". Similar to our work in Chapter 4, we transfer to unseen individuals after pretraining, but we also transfer to entirely new supervised tasks and even new sensor configurations. After self-supervised pretraining, we freeze the parameters of our model and fine-tune just two additional layers: a 16-unit hidden layer and an output layer for the supervised classification. With a challenging leave-one-patient-out evaluation that eliminates information leakage between pretraining and fine-tuning, we show that this method approaches state-of-the-art performance on three separate activities related to speech production for all seven individuals in our dataset. The transfer learning contributions of these chapters suports this dissertations adaptability objectives.

Goal 3: Enable adaptation from unlabeled data with self-supervised pretraining

Section 5.2's brain2vec model also addresses our third objective with its introduction of a self-supervised method of pretraining that does not require labeled data. Prior work relies on carefully curated labeled datasets in order to support training of supervised classification pipelines. In contrast, by pretraining with self-supervised learning at the sensor-level, with a sensor-location-aware positional embedding, brain2vec can be trained without labels, on any number of individuals, with any number of sensors in any location. To the our knowledge, this work is one of the earliest self-supervised pretraining methodologies in field of BCI for HAR that achieves worthwhile classification performance after transfering from pretraining using unlabeled data. Our self-supervised contribution supports this dissertation's objectives of improved adaptability as well as trustworthiness through improved usability of reduced dependence on labeled data.

Goal 4: Characterize neural representations and the privacy risks of individuals contributing data

Results in pursuit of our fourth goal are provided in Section 5.3. Considering the novelty of our brain2vec methodology, we aimed to explore the impact of varying pretraining cohort size, hyperparameters, and architecture design decisions in order to better understand what aspects correlate with performance. To accomplish this, we perform a large grid-search of brain2vec configurations, showing that the output dimensions of the feature extractor have a large impact on self-supervised pretraining objectives. Similar to prior work in the visual domain, we find that the pretraining performance is not consistently correlated with downstream task performance. We also show that disentangling the temporal and spatial dimensions of the positional embedding appears to make pretraining more difficult, but this potential regularization does not appear to improve down-stream classification performance. However, results indicate that larger pretraining cohorts of six participants vs. single participants may reduce the variance of down-stream performance.

We also develop two experiments designed to help assess the privacy risk faced by individuals contributing neural data to brain2vec's pretraining. Prior work investigating the privacy of HAR solutions have been limited, with none investigating self-supervised feature extractors like brain2vec. We argue that as BCI solutions become more common for everyday HAR solutions, researchers must work now to understand the privacy risks of new methods. To that end, we show that users can be re-identified with high-accuracy by simply fine-tuning on participant identifiers as a down-stream classification task. This suggests that brain2vec has potential as both a worthwhile solution for user-tracking as well as a tool for malicious re-identification of individuals contributing pretraining data.

We also contribute a novel membership inference experiment that aims to study the potential of person-level membership inference, rather than sample-level. Prior work in other domains outside of HAR are not necessarily associated with an individual, thus we introduced an experiment that uses 2-particpant brain2vec models to conduct a shadow modeling attack on pretraining member privacy for brain2vec. Our membership inference experiment does not evidence significant risk, though we discover that the contrastive loss values used during pretraining may pose a risk due their membership detection accuracy increasing above baseline levels. However, more research and development is needed to provide a more conclusive result. These contributions support this dissertations objective of building trustworthy ML approaches to HAR from bioelectric signals by investigating the privacy risks of our methods.

6.2 Future Work

Recent ML techniques have benefited from large, easily accessible datasets. For example, breakthrough work in image analysis was a combination of both new ideas and easy access to large databases of images voluntarily shared by millions of users on the internet [272]. Because individuals have their own desires to take and share photos, using their own commodity devices, the data necessary to build these solutions became abundant. Paired with advances in compute technology (i.e., Graphics Processing Unit (GPU)s), researchers were able to leverage this abundant data to further develop the capabilities of large scale ML techniques.

Researchers and practitioners are now challenged to not only consider how to enable trust in their solutions, but also how to apply these solutions to other domains that do not share the same scale of data as images or text modalities. A naive assumption is to assume that we must simply wait for the same abundance of data to emerge in these new domains. For example, expecting groundbreaking BCI solutions only once sufficient number of people are collecting data from their

BCIs and making it widely available. This is unlikely - there is no clear motivation for a user to record and share their BCI recordings in the same way users are motivated to take photos or write text. The modalities of images and text are the natural way that humans communicate. Without intervention to collect large scale data, most bioelectric-based HAR solutions such as BCI will struggle to find further success, and without success, it is unlikely to attract users. It becomes a problem of dependencies - i.e., is the cart before the horse.

Instead, like some researchers have been proposing [273], ML research efforts need to shift towards "small data" problems and their solutions. Challenges of too few samples plague methods that require direct optimization of many parameters - a well known problem in data-centric ML methods. As ML methods become more standardized, and pretraining methods produce worthwhile pretrained foundation models across domains, its believed that even small amounts of high-quality data can produce worthwhile results.

In many cases, approaches to HAR challenges avoid optimizing across users and their data, further limiting available data for training. These decisions only exacerbate issues of limited data in an attempt to manage the distribution discrepancies that occur across individuals contributing data. If we look to more prevalent domains, such as text and image analysis, leveraging unlabeled HAR datasets to pretrain models for few-shot application to downstream tasks is likely a fruitful direction. While our brain2vec contributions were notable, our grid search experiments suggest that more investigation is needed to fully understand these types of approaches.

Taking these challenges to their extreme arrives at the research efforts on incremental and continual learning. In these paradigms, samples may be few and the tasks themselves in need of algorithmic discovery. We expect HAR to be difficult to approach with continual learning due it's inherit challenges (see Section 2.2.2) and lack of clear prior work demonstrating how to transfer across distribution discrepancies. We believe that our contributions have helped move the field forward with respect to these challenges, but more work is needed. Still, it may be breakthroughs in incremental learning that instead unlock better transfer learning for the HAR domain and its problems. Therefore, we encourage researchers in the field of incremental learning to consider applying their techniques to HAR related problems.

Incremental learning is also compelling because of its interest in minimizing the number of samples that a model must store to maintain performance. Future methods that don't require recording and storing user data for HAR transfer learning will bolster security and therefore improve trust in certain scenarios. This information security is especially important in information-rich HAR scenarios that leverage potentially sensitive information. For example, it's not clear what value high-resolution brain recording may have - future algorithms may be able to extract or predict much more sensitive information than they currently can today. The future risk of today's data can be hard to understand - researchers and scientist must work now to supply future practitioners with worthwhile solutions.

Appendix A

ABBREVIATIONS

AFC Agglomerative Feature Clustering

AI Artificial Intelligence

AML Adversarial Machine Learning

BCI Brain-Computer Interface

CNN Convolutional Neural Network

CSP Common Spatial Patterns

CT Computerized Tomography

DBS Deep Brain Stimulation

DL Deep Learning

DUA Dual User-Adaption

ECoG electrocorticography

EEG Electroencepahlography

EMG Myoelectrocortography

EP Evoked Potential

ET Essential Tremor

FFT Fast Fourier Transform

FIR Finite Impulse Response

fMRI Functional Magnetic Resonance Imaging

GPI Globus Pallidus Interna

GPU Graphics Processing Unit

GB Gradient Boosting

HAR Human Activity Recognition

HCI Human Computer Interaction

HLEG High-Level Expert Group

HMI Human Machine Interaction

HRI Human Robot Interaction

ICA Independent Component Analysis

IEEG Intracranial Electroencephalography

IoT Internet-of-Things

KNN K-Nearest Neighbors

LDA Linear Discriminant Analysis

LR Logistic Regression

LReLU Leaky Rectified Linear Units

MI Mutual Information

ML Machine Learning

MRI Magnetic Resonance Imaging

NIST National Institute of Standards and Technology

NLP Natural Language Processing

OECD Organisation for Economic Co-operation and Development

PAC Phase Amplitude Coupling

PD Parkinson's Disease

PCA Principle Component Analysis

PET Positron Emission Tomography

QEEG Quantitative Electroencepahlography

RAS Right Anterior Superior

RBF Radial Basis Function

ReLU Rectified Linear Units

RF Random Forest

RNN Recurrent Neural Network

sEEG Stereotactic Electroencephalography

SSL Self-supervised Learning

STN Subthalamic Nucleus

SVM Support Vector Machine

TRB Trust-Related Behavior

VIM Ventral Intermedius Nucleus

XAI Explainable Artificial Intelligence

Bibliography

[1] Ethem Alpaydin. *Introduction to machine learning*. en. 2nd ed. Adaptive computation and machine learning. OCLC: ocn317698631. Cambridge, Mass: MIT Press, 2010. ISBN: 978-0-262-01243-0.

[2] Jason Bell. *Machine learning: hands-on for developers and technical professionals*. en. Indianapolis, Indiana: Wiley, 2015. ISBN: 978-1-118-88906-0.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[4] Ama Simons et al. "Impact of Physiological Sensor Variance on Machine Learning Algorithms". In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. ISSN: 2577-1655. Oct. 2020, pp. 241–247. DOI: 10.1109/SMC42975.2020.9282912.

[5] Kaixuan Chen et al. "Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities". en. In: *ACM Computing Surveys* 54.4 (May 2022), pp. 1–40. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3447744. URL: https://dl.acm.org/doi/10.1145/3447744 (visited on 06/22/2022).

[6] E. Ramanujam, Thinagaran Perumal, and S. Padmavathi. "Human Activity Recognition With Smartphone and Wearable Sensors Using Deep Learning Techniques: A Review". In: *IEEE Sensors Journal* 21.12 (June 2021). Conference Name: IEEE Sensors Journal, pp. 13029–13040. ISSN: 1558-1748. DOI: 10.1109/JSEN.2021.3069927.

[7] Zehua Sun et al. "Human Action Recognition From Various Data Modalities: A Review". In: *IEEE transactions on pattern analysis and machine intelligence* PP (June 2022). DOI: 10.1109/TPAMI.2022.3183112.

[8] Sizhen Bian et al. "The State-of-the-Art Sensing Techniques in Human Activity Recognition: A Survey". en. In: *Sensors* 22.12 (June 2022). 12 citations (Crossref) [2024-03-16], p. 4596. ISSN: 1424-8220. DOI: 10.3390/s22124596. URL: https://www.mdpi.com/1424-8220/22/12/4596 (visited on 03/10/2024).

[9] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". en. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (Oct. 2010), pp. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191. URL: http://ieeexplore.ieee.org/document/5288526/ (visited on 11/03/2022).

[10] Cynthia Dwork. "Differential Privacy: A Survey of Results". en. In: *Theory and Applications of Models of Computation*. Ed. by Manindra Agrawal et al. Vol. 4978. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19. ISBN: 978-3-540-79227-7 978-3-540-79228-4. DOI: 10.1007/978-3-540-79228-4_1. URL: http://link.springer.com/10.1007/978-3-540-79228-4_1 (visited on 11/20/2022).

[11] Di Liu et al. "Bringing AI to edge: From deep learning's perspective". en. In: *Neurocomputing* 485 (May 2022), pp. 297–320. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.04.141. URL: https://www.sciencedirect.com/science/article/pii/S0925231221016428 (visited on 05/28/2022).

[12] Laura von Rueden et al. "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2021). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 1–1. ISSN: 1558-2191. DOI: 10.1109/TKDE.2021.3079836.

[13] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (2019). arXiv: 1810.04805 [cs].

[14] Tom B. Brown et al. "Language Models Are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (2020). arXiv: 2005.14165 [cs].

[15] Haochen Liu et al. *Trustworthy AI: A Computational Perspective.* Number: arXiv:2107.06641 arXiv:2107.06641 [cs]. Aug. 2021. URL: http://arxiv.org/abs/2107.06641 (visited on 08/20/2022).

[16] Morgan Stuart and Milos Manic. "Survey of progress in deep neural networks for resource-constrained applications". en. In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society.* Beijing: IEEE, Oct. 2017, pp. 7259–7266. ISBN: 978-1-5386-1127-2. DOI: 10.1109/IECON.2017.8217271. URL: http://ieeexplore.ieee.org/document/8217271/ (visited on 11/19/2022).

[17] Matthias Braun, Hannah Bleher, and Patrik Hummel. "A Leap of Faith: Is There a Formula for "Trustworthy" AI?" en. In: *Hastings Center Report* 51.3 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hast.1207, pp. 17–22. ISSN: 1552-146X. DOI: 10.1002/hast.1207. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hast.1207 (visited on 09/13/2022).

[18] Martin Strobel and Reza Shokri. "Data Privacy and Trustworthy Machine Learning". In: *IEEE Security & Privacy* (2022). Conference Name: IEEE Security & Privacy, pp. 2–7. ISSN: 1558-4046. DOI: 10.1109/MSEC.2022.3178187.

[19] Cynthia Dwork. "An Ad Omnia Approach to Defining and Achieving Private Data Analysis". en. In: *Privacy, Security, and Trust in KDD.* Ed. by Francesco Bonchi et al. Vol. 4890. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–13. ISBN: 978-3-540-78477-7 978-3-540-78478-4. DOI: 10.1007/978-3-540-78478-4_1. URL: http://link.springer.com/10.1007/978-3-540-78478-4_1 (visited on 08/15/2022).

[20] Muneeb Ahmed Sahi et al. "Privacy Preservation in e-Healthcare Environments: State of the Art and Future Directions". In: *IEEE Access* 6 (2018). Conference Name: IEEE Access, pp. 464–478. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2767561.

[21] Mohamed Elhoseny et al. "Security and Privacy Issues in Medical Internet of Things: Overview, Countermeasures, Challenges and Future Directions". en. In: *Sustainability* 13.21 (Oct. 2021), p. 11645. ISSN: 2071-1050. DOI: 10.3390/su132111645. URL: https://www.mdpi.com/2071-1050/13/21/11645 (visited on 08/13/2022).

[22] Bo Li et al. *Trustworthy AI: From Principles to Practices.* Number: arXiv:2110.01167 arXiv:2110.01167 [cs]. May 2022. URL: http://arxiv.org/abs/2110.01167 (visited on 08/20/2022).

[23] Joelle Pineau et al. "Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program)". In: *Journal of Machine Learning Research* 22.164 (2021), pp. 1–20.

[24] Arik Friedman and Assaf Schuster. "Data mining with differential privacy". en. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*. Washington, DC, USA: ACM Press, 2010, p. 493. ISBN: 978-1-4503-0055-1. DOI: 10.1145/1835804.1835868. URL: http://dl.acm.org/citation.cfm?doid=1835804.1835868 (visited on 11/20/2022).

[25] Reza Shokri et al. *Membership Inference Attacks against Machine Learning Models*. 2181 citations (Semantic Scholar/arXiv) [2023-02-26] arXiv:1610.05820 [cs, stat]. Mar. 2017. URL: http://arxiv.org/abs/1610.05820 (visited on 11/20/2022).

[26] Christopher A. Choquette-Choo et al. "Label-Only Membership Inference Attacks". en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2021, pp. 1964–1974. URL: https://proceedings.mlr.press/v139/choquette-choo21a.html (visited on 11/20/2022).

[27] Jiayuan Ye et al. *Enhanced Membership Inference Attacks against Machine Learning Models*. 30 citations (Semantic Scholar/arXiv) [2023-02-26] arXiv:2111.09679 [cs, stat]. Sept. 2022. URL: http://arxiv.org/abs/2111.09679 (visited on 11/20/2022).

[28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". en. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 02/08/2020).

[29] Henry Friday Nweke et al. "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges". en. In: *Expert Systems with Applications* 105 (Sept. 2018), pp. 233–261. ISSN: 09574174. DOI: 10.1016/j.eswa.2018.03.056. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417418302136 (visited on 06/22/2022).

[30] Ming Zeng et al. "Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors". en. In: ICST, 2014. ISBN: 978-1-63190-024-2. DOI: 10.4108/icst.mobicase.2014.257786. URL: http://eudl.eu/doi/10.4108/icst.mobicase.2014.257786 (visited on 04/22/2018).

[31] Jianbo Yang et al. "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition." In: *IJCAI*. 2015, pp. 3995–4001.

[32] Ki-Hee Park and Seong-Whan Lee. "Movement intention decoding based on deep learning for multiuser myoelectric interfaces". In: *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*. Feb. 2016, pp. 1–2. DOI: 10.1109/IWW-BCI.2016.7457459.

[33] Thomas Plötz and Yu Guan. "Deep learning for human activity recognition in mobile computing". In: *Computer* 51.5 (2018), pp. 50–59.

[34] Francisco Javier Ordóñez Morales and Daniel Roggen. "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations". en. In: *Proceedings of the 2016 ACM International Symposium on Wearable Computers - ISWC '16*. Heidelberg, Germany: ACM Press, 2016, pp. 92–99. ISBN: 978-1-4503-4460-9. DOI: 10.1145/2971763.2971764. URL: http://dl.acm.org/citation.cfm?doid=2971763.2971764 (visited on 02/08/2020).

[35] Francisco Javier Ordóñez and Daniel Roggen. "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition". en. In: *Sensors* 16.1 (Jan. 2016), p. 115. DOI: 10.3390/s16010115. URL: http://www.mdpi.com/1424-8220/16/1/115 (visited on 04/01/2018).

[36] Yu Hu et al. "A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition". en. In: *PLOS ONE* 13.10 (Oct. 2018), e0206049. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0206049. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0206049 (visited on 01/16/2020).

[37] Adam Hartwell, Visakan Kadirkamanathan, and Sean R. Anderson. "Compact Deep Neural Networks for Computationally Efficient Gesture Classification From Electromyography Signals". In: *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. ISSN: 2155-1774. Aug. 2018, pp. 891–896. DOI: 10.1109/BIOROB.2018.8487853.

[38] Daniele Ravi et al. "Deep learning for human activity recognition: A resource efficient implementation on low-power devices". In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. ISSN: 2376-8894. June 2016, pp. 71–76. DOI: 10.1109/BSN.2016.7516235.

[39] Morgan Stuart et al. "Machine Learning for Deep Brain Stimulation Efficacy using Dense Array EEG". en. In: *2019 12th International Conference on Human System Interaction (HSI)*. Richmond, VA, USA: IEEE, June 2019, pp. 143–150. ISBN: 978-1-72813-980-7. DOI: 10.1109/HSI47298.2019.8942619. URL: https://ieeexplore.ieee.org/document/8942619/ (visited on 11/19/2022).

[40] Morgan Stuart et al. "An Interpretable Deep Learning Model for Speech Activity Detection Using Electrocorticographic Signals". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30 (2022), pp. 2783–2792.

[41] Morgan Stuart and Milos Manic. "Deep Learning Shared Bandpass Filters for Resource-Constrained Human Activity Recognition". In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 39089–39097. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3064031.

[42] Srdjan Lesaja et al. "Self-Supervised Learning of Neural Speech Representations From Unlabeled Intracranial Signals". In: *IEEE Access* 10 (2022). Conference Name: IEEE Access, pp. 133526–133538. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3230688.

[43] Brett W. Israelsen and Nisar R. Ahmed. ""Dave...I can assure you ...that it's going to be all right ..." A Definition, Case for, and Survey of Algorithmic Assurances in Human-Autonomy Trust Relationships". en. In: *ACM Computing Surveys* 51.6 (Nov. 2019), pp. 1–37. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3267338. URL: https://dl.acm.org/doi/10.1145/3267338 (visited on 07/15/2023).

[44] Jeannette M. Wing. "Trustworthy AI". en. In: *Communications of the ACM* 64.10 (Oct. 2021), pp. 64–71. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3448248. URL: https://dl.acm.org/doi/10.1145/3448248 (visited on 08/20/2022).

[45] Fred B. Schneider and National Research Council (U.S.), eds. *Trust in cyberspace*. en. Washington, D.C: National Academy Press, 1999. ISBN: 978-0-309-06558-0.

[46] Gary Gensler and Lily Bailey. "Deep Learning and Financial Stability". en. In: *SSRN Electronic Journal* (2020). ISSN: 1556-5068. DOI: 10.2139/ssrn.3723132. URL: https://www.ssrn.com/abstract=3723132 (visited on 08/08/2023).

[47] Ekaterina Svetlova. "AI ethics and systemic risks in finance". en. In: *AI and Ethics* 2.4 (Nov. 2022), pp. 713–725. ISSN: 2730-5953, 2730-5961. DOI: 10.1007/s43681-021-00129-1. URL: https://link.springer.com/10.1007/s43681-021-00129-1 (visited on 08/08/2023).

[48] Dwight Horne. *PwnPilot: Reflections on Trusting Trust in the Age of Large Language Models and AI Code Assistants.* July 2023.

[49] Paweł Niszczota and Paul Conway. *Judgments of research co-created by generative AI: experimental evidence.* Publication Title: arXiv e-prints ADS Bibcode: 2023arXiv230511873N. May 2023. DOI: 10.48550/arXiv.2305.11873. URL: https://ui.adsabs.harvard.edu/abs/2023arXiv230511873N (visited on 08/09/2023).

[50] Philipp Hacker, Andreas Engel, and Marco Mauer. "Regulating ChatGPT and other Large Generative AI Models". en. In: *2023 ACM Conference on Fairness, Accountability, and Transparency*. Chicago IL USA: ACM, June 2023, pp. 1112–1123. ISBN: 9798400701924. DOI: 10.1145/3593013.3594067. URL: https://dl.acm.org/doi/10.1145/3593013.3594067 (visited on 08/09/2023).

[51] John D. Lee and Katrina A. See. "Trust in Automation: Designing for Appropriate Reliance". en. In: *Human Factors* 46.1 (Mar. 2004). Publisher: SAGE Publications Inc, pp. 50–80. ISSN: 0018-7208. DOI: 10.1518/hfes.46.1.50_30392. URL: https://journals.sagepub.com/doi/abs/10.1518/hfes.46.1.50_30392 (visited on 11/15/2022).

[52] Miles Brundage et al. *Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims.* en. Number: arXiv:2004.07213 arXiv:2004.07213 [cs]. Apr. 2020. URL: http://arxiv.org/abs/2004.07213 (visited on 08/20/2022).

[53] Centers for Medicare & Medicaid Services. *The Health Insurance Portability and Accountability Act of 1996 (HIPAA).* Online at http://www.cms.hhs.gov/hipaa/. 1996.

[54] *2018 reform of EU data protection rules.* European Commission. May 25, 2018. URL: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf (visited on 06/17/2019).

[55] European Commission. *Ethics Guidelines for Trustworthy AI.* 2019. URL: https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf (visited on 10/19/2022).

[56] Luciano Floridi. "Establishing the rules for building trustworthy AI". en. In: *Nature Machine Intelligence* 1.6 (June 2019), pp. 261–262. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0055-y. URL: http://www.nature.com/articles/s42256-019-0055-y (visited on 08/20/2022).

[57] *The OECD Artificial Intelligence (AI) Principles.* en. URL: https://oecd.ai/en/ai-principles (visited on 10/08/2022).

[58] *Blueprint for an AI Bill of Rights.* en-US. URL: https://www.whitehouse.gov/ostp/ai-bill-of-rights/ (visited on 10/08/2022).

[59] Filip Došilović, Mario Brcic, and Nikica Hlupic. *Explainable Artificial Intelligence: A Survey.* May 2018. DOI: 10.23919/MIPRO.2018.8400040.

[60] Leilani H. Gilpin et al. *Explaining Explanations: An Overview of Interpretability of Machine Learning.* en. arXiv:1806.00069 [cs, stat]. Feb. 2019. URL: http://arxiv.org/abs/1806.00069 (visited on 07/15/2023).

[61] Adrian Weller. "Transparency: Motivations and Challenges". en. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 23–40. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_2. URL: https://doi.org/10.1007/978-3-030-28954-6_2 (visited on 07/26/2022).

[62] Ričards Marcinkevičs and Julia E. Vogt. *Interpretability and Explainability: A Machine Learning Zoo Mini-tour*. arXiv:2012.01805 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2012.01805. URL: http://arxiv.org/abs/2012.01805 (visited on 03/17/2024).

[63] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". en. In: *Nature Machine Intelligence* 1.5 (May 2019), pp. 206–215. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0048-x. URL: https://www.nature.com/articles/s42256-019-0048-x (visited on 06/02/2023).

[64] Finale Doshi-Velez and Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. en. arXiv:1702.08608 [cs, stat]. Mar. 2017. URL: http://arxiv.org/abs/1702.08608 (visited on 07/15/2023).

[65] Zachary C. Lipton. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." en. In: *Queue* 16.3 (June 2018), pp. 31–57. ISSN: 1542-7730, 1542-7749. DOI: 10.1145/3236386.3241340. URL: https://dl.acm.org/doi/10.1145/3236386.3241340 (visited on 07/15/2023).

[66] Emma E. Levine and Maurice E. Schweitzer. "Prosocial lies: When deception breeds trust". en. In: *Organizational Behavior and Human Decision Processes* 126 (Jan. 2015), pp. 88–106. ISSN: 07495978. DOI: 10.1016/j.obhdp.2014.10.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0749597814000983 (visited on 11/02/2022).

[67] Andrew L. Beam, Arjun K. Manrai, and Marzyeh Ghassemi. "Challenges to the Reproducibility of Machine Learning Models in Health Care". In: *JAMA* 323.4 (Jan. 2020), pp. 305–306. ISSN: 0098-7484. DOI: 10.1001/jama.2019.20866. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7335677/ (visited on 11/03/2022).

[68] Benjamin J. Heil et al. "Reproducibility standards for machine learning in the life sciences". en. In: *Nature Methods* 18.10 (Oct. 2021). Number: 10 Publisher: Nature Publishing Group, pp. 1132–1135. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01256-7. URL: https://www.nature.com/articles/s41592-021-01256-7 (visited on 11/03/2022).

[69] Samuel Warren and Louis Brandeis. "The right to privacy". In: *Columbia University Press* In Killing the Messenger: 100 Years of Media Criticism (1989), pp. 1–21.

[70] Dorothy J Glancy. "The Invention of the Right to Privacy". en. In: *Arizona Law Review* 21 (1979).

[71] Jamal Greene. "The so-called right to privacy". In: *UC Davis L. Rev.* 43 (2009). Publisher: HeinOnline, p. 715.

[72] United Nations General Assembly. *The Universal Declaration of Human Rights*. 1948.

[73] Julia Lane and Claudia Schur. "Balancing Access to Health Data and Privacy: A Review of the Issues and Approaches for the Future". en. In: *Health Services Research* 45.5p2 (2010). 46 citations (Crossref) [2023-09-04] _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.14 6773.2010.01141.x, pp. 1456–1467. ISSN: 1475-6773. DOI: 10.1111/j.1475-6773.2010.01141.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-6773.2010.01141.x (visited on 08/13/2022).

[74] B. A. Malin. “An Evaluation of the Current State of Genomic Data Privacy Protection Technology and a Roadmap for the Future”. en. In: *Journal of the American Medical Informatics Association* 12.1 (Oct. 2004). 77 citations (Crossref) [2023-09-04], pp. 28–34. ISSN: 1067-5027, 1527-974X. DOI: 10.1197/jamia.M1603. URL: https://academic.oup.com/jamia/article-lookup/doi/10.1197/jamia.M1603 (visited on 08/13/2022).

[75] Shui Yu. “Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data”. In: *IEEE Access* 4 (2016). Conference Name: IEEE Access, pp. 2751–2763. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2577036.

[76] Elham Tabassi et al. *A taxonomy and terminology of adversarial machine learning*. en. preprint. Oct. 2019. DOI: 10.6028/NIST.IR.8269-draft. URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8269-draft.pdf (visited on 09/04/2023).

[77] Hongyan Chang and Reza Shokri. “On the Privacy Risks of Algorithmic Fairness”. In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. Sept. 2021, pp. 292–303. DOI: 10.1109/EuroSP51992.2021.00028.

[78] Fakhreddine Karray et al. “Human-Computer Interaction: Overview on State of the Art”. en. In: *International Journal on Smart Sensing and Intelligent Systems* 1.1 (Jan. 2008), pp. 137–159. ISSN: 1178-5608. DOI: 10.21307/ijssis-2017-283. URL: https://www.sciendo.com/article/10.21307/ijssis-2017-283 (visited on 11/27/2022).

[79] Gong Chao. “Human-Computer Interaction: Process and Principles of Human-Computer Interface Design”. In: *2009 International Conference on Computer and Automation Engineering*. 40 citations (Semantic Scholar/DOI) [2023-08-19] 20 citations (Crossref) [2023-08-19]. Mar. 2009, pp. 230–233. DOI: 10.1109/ICCAE.2009.23.

[80] Gaurav Sinha, Rahul Shahi, and Mani Shankar. “Human Computer Interaction”. In: *2010 3rd International Conference on Emerging Trends in Engineering and Technology*. 15 citations (Semantic Scholar/DOI) [2023-08-19] 18 citations (Crossref) [2023-08-19] ISSN: 2157-0485. Nov. 2010, pp. 1–4. DOI: 10.1109/ICETET.2010.85.

[81] Donald A. Norman. “Stages and levels in human-machine interaction”. en. In: *International Journal of Man-Machine Studies* 21.4 (Oct. 1984), pp. 365–375. ISSN: 00207373. DOI: 10.1016/S0020-7373(84)80054-1. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020737384800541 (visited on 11/27/2022).

[82] Ewart J. De Visser, Richard Pak, and Tyler H. Shaw. “From ‘automation’ to ‘autonomy’: the importance of trust repair in human–machine interaction”. en. In: *Ergonomics* 61.10 (Oct. 2018), pp. 1409–1427. ISSN: 0014-0139, 1366-5847. DOI: 10.1080/00140139.2018.1457725. URL: https://www.tandfonline.com/doi/full/10.1080/00140139.2018.1457725 (visited on 08/06/2023).

[83] Jayavardhana Gubbi et al. “Internet of Things (IoT): A vision, architectural elements, and future directions”. en. In: *Future Generation Computer Systems* 29.7 (Sept. 2013), pp. 1645–1660. ISSN: 0167739X. DOI: 10.1016/j.future.2013.01.010. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167739X13000241 (visited on 09/05/2018).

[84] R. Benjamin Knapp, Jonghwa Kim, and Elisabeth André. “Physiological Signals and Their Use in Augmenting Emotion Recognition for Human–Machine Interaction”. en. In: *Emotion-Oriented Systems*. Ed. by Roddy Cowie, Catherine Pelachaud, and Paolo Petta. Series Title: Cognitive Technologies. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 133–159. ISBN: 978-3-642-15183-5 978-3-642-15184-2. DOI: 10.1007/978-

3-642-15184-2_9. URL: http://link.springer.com/10.1007/978-3-642-15184-2_9 (visited on 08/06/2023).

[85] Ruiyang Yin et al. “Wearable Sensors-Enabled Human–Machine Interaction Systems: From Design to Application”. en. In: *Advanced Functional Materials* 31.11 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adfm.202008936, p. 2008936. ISSN: 1616-3028. DOI: 10.1002/adfm.202008936. URL: http://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202008936 (visited on 11/27/2022).

[86] Monica Tiboni et al. “Sensors and Actuation Technologies in Exoskeletons: A Review”. In: *Sensors (Basel, Switzerland)* 22.3 (Jan. 2022), p. 884. ISSN: 1424-8220. DOI: 10.3390/s22030884. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8839165/ (visited on 11/27/2022).

[87] Resul Das et al. “A Survey on the Internet of Things Solutions for the Elderly and Disabled: Applications, Prospects, and Challenges”. en. In: *International Journal of Computer Networks And Applications* 4.3 (June 2017), p. 1. ISSN: 2395-0455. DOI: 10.22247/ijcna/2017/49023. URL: http://www.ijcna.org/Manuscripts/IJCNA-2017-O-08.pdf (visited on 02/08/2020).

[88] Emiro De-La-Hoz-Franco et al. “Sensor-Based Datasets for Human Activity Recognition – A Systematic Review of Literature”. In: *IEEE Access* 6 (2018), pp. 59192–59210. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2873502.

[89] Morgan Stuart et al. “Machine learning for deep brain stimulation efficacy using dense array eeg”. In: *2019 12th International Conference on Human System Interaction (HSI)*. IEEE. 2019, pp. 143–150.

[90] *Common Voice*. URL: https://commonvoice.mozilla.org/en (visited on 11/19/2022).

[91] Ju Ren et al. “Edge Computing for the Internet of Things”. In: *IEEE Network* 32.1 (Jan. 2018), pp. 6–7. ISSN: 1558-156X. DOI: 10.1109/MNET.2018.8270624.

[92] Robert S. Witte and John S. Witte. *Statistics*. en. Eleventh edition. Hoboken, NJ: Wiley, 2017. ISBN: 978-1-119-25451-5.

[93] Yves Kodratoff. *Introduction to machine learning*. en. OCLC: 915911871. 2014.

[94] N. Swann et al. “Deep Brain Stimulation of the Subthalamic Nucleus Alters the Cortical Profile of Response Inhibition in the Beta Frequency Band: A Scalp EEG Study in Parkinson’s Disease”. en. In: *Journal of Neuroscience* 31.15 (Apr. 2011), pp. 5721–5729. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.6135-10.2011. URL: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.6135-10.2011 (visited on 12/21/2018).

[95] Nicole C. Swann et al. “Elevated Synchrony in Parkinson’s Disease Detected with Electroencephalography”. In: *Annals of neurology* 78.5 (Nov. 2015), pp. 742–750. ISSN: 0364-5134. DOI: 10.1002/ana.24507. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4623949/ (visited on 06/09/2019).

[96] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[97] Weibo Liu et al. “A survey of deep neural network architectures and their applications”. In: *Neurocomputing* 234 (2017), pp. 11–26.

[98] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision". In: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE. 2010, pp. 253–256.

[99] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Aistats*. Vol. 9. 2010, pp. 249–256.

[100] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Cognitive modeling* 5.3 (1988), p. 1.

[101] Hehe Fan, Yi Yang, and Mohan Kankanhalli. "Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, 2021, pp. 14199–14208. ISBN: 978-1-66544-509-2. DOI: 10.1109/CVPR46437.2021.01398.

[102] Hassan Akbari et al. "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24206–24221.

[103] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is Space-Time Attention All You Need for Video Understanding?" In: *arXiv:2102.05095 [cs]* (2021). arXiv: 2102.05095 [cs].

[104] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *arXiv:2005.12872 [cs]* (2020). arXiv: 2005.12872 [cs].

[105] Anurag Arnab et al. "ViViT: A Video Vision Transformer". In: *arXiv:2103.15691 [cs]* (2021). arXiv: 2103.15691 [cs].

[106] Hengshuang Zhao et al. "Point transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268.

[107] Jay Alammar. *The Illustrated Transformer*. https://jalammar.github.io/illustrated-transformer/. Accessed: 2022-10-02.

[108] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762 [cs]* (2017). arXiv: 1706.03762 [cs].

[109] Laura von Rueden et al. "Informed Machine Learning – A Taxonomy and Survey of Integrating Knowledge into Learning Systems". In: *arXiv:1903.12394 [cs, stat]* (Feb. 2020). arXiv: 1903.12394. URL: http://arxiv.org/abs/1903.12394 (visited on 10/24/2020).

[110] George Karniadakis et al. "Physics-informed machine learning". In: (May 2021), pp. 1–19. DOI: 10.1038/s42254-021-00314-5.

[111] Ribana Roscher et al. "Explainable Machine Learning for Scientific Insights and Discoveries". en. In: *IEEE Access* 8 (2020), pp. 42200–42216. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2976199. URL: https://ieeexplore.ieee.org/document/9007737/ (visited on 10/23/2023).

[112] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (Jan. 2021). Conference Name: Proceedings of the IEEE, pp. 43–76. ISSN: 1558-2256. DOI: 10.1109/JPROC.2020.3004555.

[113] Xingyi Yang et al. *Transfer Learning or Self-supervised Learning? A Tale of Two Pretraining Paradigms*. Number: arXiv:2007.04234 arXiv:2007.04234 [cs, stat]. June 2020. URL: http://arxiv.org/abs/2007.04234 (visited on 08/04/2022).

[114] Rajat Raina et al. "Self-taught learning: transfer learning from unlabeled data". In: *Proceedings of the 24th international conference on Machine learning*. ICML '07. New York, NY, USA: Association for Computing Machinery, June 2007, pp. 759–766. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273592. URL: http://doi.org/10.1145/1273496.1273592 (visited on 11/03/2022).

[115] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. "The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects". In: *Frontiers in Psychology* 4 (Aug. 2013), p. 504. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00504. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3732997/ (visited on 08/09/2023).

[116] Matthias De Lange et al. "A continual learning survey: Defying forgetting in classification tasks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). arXiv:1909.08383 [cs, stat], pp. 1–1. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3057446. URL: http://arxiv.org/abs/1909.08383 (visited on 08/09/2023).

[117] Michael McCloskey and Neal J. Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". en. In: *Psychology of Learning and Motivation*. Vol. 24. Elsevier, 1989, pp. 109–165. ISBN: 978-0-12-543324-2. DOI: 10.1016/S0079-7421(08)60536-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0079742108605368 (visited on 08/28/2023).

[118] Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Vol. 1. Springer, 2018.

[119] Matthias De Lange et al. "Unsupervised Model Personalization While Preserving Privacy and Scalability: An Open Problem". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 14451–14460. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.01447. URL: https://ieeexplore.ieee.org/document/9157789/ (visited on 09/17/2023).

[120] Jiahua Dong et al. "Federated Class-Incremental Learning". en. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 10154–10163. ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.00992. URL: https://ieeexplore.ieee.org/document/9878590/ (visited on 09/17/2023).

[121] P. Samarati and L. Sweeney. "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression". In: *IEEE Symposium on Security and Privacy*. 1998. URL: https://www.semanticscholar.org/paper/Protecting-privacy-when-disclosing-information%3A-and-Samarati-Sweeney/7df12c498fecedac4ab6034d3a8 (visited on 09/09/2023).

[122] Ashwin Machanavajjhala et al. "L-diversity: Privacy beyond k-anonymity". In: *ACM Transactions on Knowledge Discovery from Data* 1.1 (Mar. 2007), 3–es. ISSN: 1556-4681. DOI: 10.1145/1217299.1217302. URL: https://doi.org/10.1145/1217299.1217302 (visited on 09/09/2023).

[123] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. "Closeness: A New Privacy Measure for Data Publishing". en. In: *IEEE Transactions on Knowledge and Data Engineering* 22.7 (July 2010), pp. 943–956. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.139. URL: http://ieeexplore.ieee.org/document/5072216/ (visited on 09/09/2023).

[124] Nils Homer et al. "Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays". en. In: *PLoS Genetics* 4.8 (Aug. 2008). Ed. by Peter M. Visscher, e1000167. ISSN: 1553-7404. DOI: 10.1371/journal.pgen.1000167. URL: https://dx.plos.org/10.1371/journal.pgen.1000167 (visited on 03/17/2024).

[125] Michael Backes et al. "Membership Privacy in MicroRNA-based Studies". en. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Vienna Austria: ACM, Oct. 2016, pp. 319–330. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978355. URL: https://dl.acm.org/doi/10.1145/2976749.2978355 (visited on 03/17/2024).

[126] Nicholas Carlini et al. "Membership Inference Attacks From First Principles". en. In: *2022 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2022, pp. 1897–1914. ISBN: 978-1-66541-316-9. DOI: 10.1109/SP46214.2022.9833649. URL: https://ieeexplore.ieee.org/document/9833649/ (visited on 02/25/2024).

[127] Frances M. Weaver et al. "Randomized trial of deep brain stimulation for Parkinson disease". In: *Neurology* 79.1 (July 2012), pp. 55–65. ISSN: 0028-3878. DOI: 10.1212/WNL.0b013e31825dcdc1. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3385495/ (visited on 06/12/2019).

[128] Kenneth A. Follett et al. "Pallidal versus Subthalamic Deep-Brain Stimulation for Parkinson's Disease". In: *New England Journal of Medicine* 362.22 (June 2010), pp. 2077–2091. ISSN: 0028-4793. DOI: 10.1056/NEJMoa0907083. URL: https://doi.org/10.1056/NEJMoa0907083 (visited on 06/11/2019).

[129] Frances M Weaver et al. "Bilateral deep brain stimulation vs best medical therapy for patients with advanced Parkinson disease: a randomized controlled trial". In: *Jama* 301.1 (2009), pp. 63–73.

[130] Thomas Wichmann and Mahlon R. DeLong. "Deep Brain Stimulation for Neurologic and Neuropsychiatric Disorders". en. In: *Neuron* 52.1 (Oct. 2006), pp. 197–204. ISSN: 08966273. DOI: 10.1016/j.neuron.2006.09.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S089662730600729X (visited on 06/11/2019).

[131] P. Limousin et al. "Effect of parkinsonian signs and symptoms of bilateral subthalamic nucleus stimulation." English. In: *Lancet (London, England)* 345.8942 (1995), pp. 91–95. ISSN: 0140-6736. URL: http://search.proquest.com/docview/77127288?rfr_id=info%3Axri%2Fsid%3Aprimo (visited on 06/11/2019).

[132] Deep-Brain Stimulation for Parkinson's Disease Study Group. "Deep-brain stimulation of the subthalamic nucleus or the pars interna of the globus pallidus in Parkinson's disease". In: *New England Journal of Medicine* 345.13 (2001), pp. 956–963.

[133] J. F. Baizabal-Carvallo et al. "The safety and efficacy of thalamic deep brain stimulation in essential tremor: 10 years and beyond". en. In: *Journal of Neurology, Neurosurgery & Psychiatry* 85.5 (May 2014), pp. 567–572. ISSN: 0022-3050. DOI: 10.1136/jnnp-2013-304943. URL: http://jnnp.bmj.com/cgi/doi/10.1136/jnnp-2013-304943 (visited on 06/11/2019).

[134] P. Richard Schuurman et al. "A Comparison of Continuous Thalamic Stimulation and Thalamotomy for Suppression of Severe Tremor". en. In: *New England Journal of Medicine* 342.7 (Feb. 2000), pp. 461–468. ISSN: 0028-4793, 1533-4406. DOI: 10.1056/NEJM200002173420703. URL: http://www.nejm.org/doi/abs/10.1056/NEJM200002173420703 (visited on 06/11/2019).

[135] Wendell Lake, Peter Hedera, and Peter Konrad. "Deep Brain Stimulation for Treatment of Tremor". en. In: *Neurosurgery Clinics of North America* 30.2 (Apr. 2019), pp. 147–159. ISSN: 10423680. DOI: 10.1016/j.nec.2019.01.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S1042368019300026 (visited on 06/11/2019).

[136] Annaelle Devergnas and Thomas Wichmann. "Cortical Potentials Evoked by Deep Brain Stimulation in the Subthalamic Area". English. In: *Frontiers in Systems Neuroscience* 5 (2011). ISSN: 1662-5137. DOI: 10.3389/fnsys.2011.00030. URL: https://www.frontiersin.org/articles/10.3389/fnsys.2011.00030/full (visited on 06/12/2019).

[137] Laura Cif et al. "Long-term follow-up of DYT1 dystonia patients treated by deep brain stimulation: An open-label study: Clinical Course of DYT1 Dystonia with DBS". en. In: *Movement Disorders* 25.3 (Feb. 2010), pp. 289–299. ISSN: 08853185. DOI: 10.1002/mds.22802. URL: http://doi.wiley.com/10.1002/mds.22802 (visited on 06/11/2019).

[138] Edward F. Chang et al. "Long-Term Benefit Sustained after Bilateral Pallidal Deep Brain Stimulation in Patients with Refractory Tardive Dystonia". In: *Stereotactic and Functional Neurosurgery* 88.5 (2010), pp. 304–310. ISSN: 1011-6125, 1423-0372. DOI: 10.1159/000316763. URL: https://www.karger.com/Article/FullText/316763 (visited on 06/11/2019).

[139] Doris D. Wang et al. "Pallidal Deep-Brain Stimulation Disrupts Pallidal Beta Oscillations and Coherence with Primary Motor Cortex in Parkinson's Disease". en. In: *The Journal of Neuroscience* 38.19 (May 2018), pp. 4556–4568. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.0431-18.2018. URL: http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0431-18.2018 (visited on 06/09/2019).

[140] Coralie de Hemptinne et al. "Therapeutic deep brain stimulation reduces cortical phase-amplitude coupling in Parkinson's disease". In: *Nature neuroscience* 18.5 (May 2015), pp. 779–786. ISSN: 1097-6256. DOI: 10.1038/nn.3997. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4414895/ (visited on 06/09/2019).

[141] Jeff M. Bronstein et al. "Deep Brain Stimulation for Parkinson Disease: An Expert Consensus and Review of Key Issues". en. In: *Archives of Neurology* 68.2 (Feb. 2011). ISSN: 0003-9942. DOI: 10.1001/archneurol.2010.260. URL: http://archneur.jamanetwork.com/article.aspx?doi=10.1001/archneurol.2010.260 (visited on 06/11/2019).

[142] Thomas Koeglsperger et al. "Deep Brain Stimulation Programming for Movement Disorders: Current Concepts and Evidence-Based Strategies". English. In: *Frontiers in Neurology* 10 (2019). ISSN: 1664-2295. DOI: 10.3389/fneur.2019.00410. URL: https://www.frontiersin.org/articles/10.3389/fneur.2019.00410/full (visited on 06/11/2019).

[143] Jens Volkmann et al. "Introduction to the programming of deep brain stimulators". en. In: *Movement Disorders* 17.S3 (2002), S181–S187. ISSN: 1531-8257. DOI: 10.1002/mds.10162. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mds.10162 (visited on 06/11/2019).

[144] Roy A. E. Bakay. "Thalamic Deep Brain Stimulation for Essential Tremor: Relation of Lead Location to Outcome". en. In: *Neurosurgery* 55.1 (July 2004), pp. 266–267. ISSN: 0148-396X. DOI: 10.1227/01.NEU.0000134764.76223.4B. URL: https://academic.oup.com/neurosurgery/article/55/1/266/2736027 (visited on 06/11/2019).

[145] Joshua K. Wong et al. "STN vs. GPi deep brain stimulation for tremor suppression in Parkinson disease: A systematic review and meta-analysis". en. In: *Parkinsonism & Related Disorders* 58 (Jan. 2019), pp. 56–62. ISSN: 13538020. DOI: 10.1016/j.parkreldis.2018.08.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S1353802018303766 (visited on 06/11/2019).

[146] Alexis M. Kuncel and Warren M. Grill. "Selection of stimulus parameters for deep brain stimulation". In: *Clinical Neurophysiology* 115.11 (Nov. 2004), pp. 2431–2441. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2004.05.031. URL: http://www.sciencedirect.com/science/article/pii/S1388245704002287 (visited on 06/11/2019).

[147] Boris Iglewicz and David C. Hoaglin. *How to detect and handle outliers.* en. ASQC basic references in quality control v. 16. Milwaukee, Wis: ASQC Quality Press, 1993. ISBN: 978-0-87389-247-6.

[148] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[149] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. "Optimal spatial filtering of single trial EEG during imagined hand movement". en. In: *IEEE Transactions on Rehabilitation Engineering* 8.4 (Dec. 2000), pp. 441–446. ISSN: 1063-6528, 1558-0024. DOI: 10.1109/86.895946. URL: https://ieeexplore.ieee.org/document/895946/ (visited on 06/11/2019).

[150] Alexandre Eusebio et al. "Resonance in subthalamo-cortical circuits in Parkinson's disease". en. In: *Brain* 132.8 (Aug. 2009), pp. 2139–2150. ISSN: 0006-8950. DOI: 10.1093/brain/awp079. URL: https://academic.oup.com/brain/article/132/8/2139/266708 (visited on 06/09/2019).

[151] Colum D. MacKinnon et al. "Stimulation through electrodes implanted near the subthalamic nucleus activates projections to motor areas of cerebral cortex in patients with Parkinson's disease". en. In: *European Journal of Neuroscience* 21.5 (2005), pp. 1394–1402. ISSN: 1460-9568. DOI: 10.1111/j.1460-9568.2005.03952.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-9568.2005.03952.x (visited on 06/09/2019).

[152] Zhen Ni et al. "Pallidal deep brain stimulation modulates cortical excitability and plasticity". en. In: *Annals of Neurology* 83.2 (2018), pp. 352–362. ISSN: 1531-8249. DOI: 10.1002/ana.25156. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ana.25156 (visited on 06/09/2019).

[153] Stephen Tisch et al. "Cortical evoked potentials from pallidal stimulation in patients with primary generalized dystonia". en. In: *Movement Disorders* 23.2 (2008), pp. 265–273. ISSN: 1531-8257. DOI: 10.1002/mds.21835. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mds.21835 (visited on 06/09/2019).

[154] Adrian W. Laxton et al. "A phase I trial of deep brain stimulation of memory circuits in Alzheimer's disease". en. In: *Annals of Neurology* 68.4 (Oct. 2010), pp. 521–534. ISSN: 03645134. DOI: 10.1002/ana.22089. URL: http://doi.wiley.com/10.1002/ana.22089 (visited on 05/17/2019).

[155] Clement Hamani et al. “Memory enhancement induced by hypothalamic/fornix deep brain stimulation”. en. In: *Annals of Neurology* 63.1 (Jan. 2008), pp. 119–123. ISSN: 03645134, 15318249. DOI: 10.1002/ana.21295. URL: http://doi.wiley.com/10.1002/ana.21295 (visited on 06/09/2019).

[156] James M. Broadway et al. “Frontal Theta Cordance Predicts 6-Month Antidepressant Response to Subcallosal Cingulate Deep Brain Stimulation for Treatment-Resistant Depression: A Pilot Study”. en. In: *Neuropsychopharmacology* 37.7 (June 2012), pp. 1764–1772. ISSN: 1740-634X. DOI: 10.1038/npp.2012.23. URL: https://www.nature.com/articles/npp201223 (visited on 06/09/2019).

[157] G. Pfurtscheller and C. Neuper. “Motor imagery and direct brain-computer communication”. In: *Proceedings of the IEEE* 89.7 (July 2001), pp. 1123–1134. ISSN: 0018-9219. DOI: 10.1109/5.939829.

[158] G. Townsend, B. Graimann, and G. Pfurtscheller. “Continuous EEG Classification During Motor Imagery—Simulation of an Asynchronous BCI”. en. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12.2 (June 2004), pp. 258–265. ISSN: 1534-4320. DOI: 10.1109/TNSRE.2004.827220. URL: http://ieeexplore.ieee.org/document/1304866/ (visited on 06/19/2019).

[159] Yijun Wang, Shangkai Gao, and Xiaornog Gao. “Common Spatial Pattern Method for Channel Selelction in Motor Imagery Based Brain-computer Interface”. In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. Shanghai, China: IEEE, 2005, pp. 5392–5395. ISBN: 978-0-7803-8741-6. DOI: 10.1109/IEMBS.2005.1615701. URL: http://ieeexplore.ieee.org/document/1615701/ (visited on 06/19/2019).

[160] Tulga Kalayci and Ozcan Ozdamar. “Wavelet preprocessing for automated neural network detection of EEG spikes”. In: *IEEE engineering in medicine and biology magazine* 14.2 (1995), pp. 160–166.

[161] Alexandros T Tzallas, Markos G Tsipouras, and Dimitrios I Fotiadis. “Epileptic Seizure Detection in Electroencephalograms using Time-Frequency Analysis”. en. In: (), p. 9.

[162] Turker Tekin Erguzel et al. “Neural Network Based Response Prediction of rTMS in Major Depressive Disorder Using QEEG Cordance”. In: *Psychiatry Investigation* 12.1 (Jan. 2015), pp. 61–65. ISSN: 1738-3684. DOI: 10.4306/pi.2015.12.1.61. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4310922/ (visited on 05/03/2019).

[163] Jacek Rumiński. “Reliability of Pulse Measurements in Videoplethysmography”. en. In: *Metrology and Measurement Systems* 23.3 (Sept. 2016), pp. 359–371. ISSN: 2300-1941. DOI: 10.1515/mms-2016-0040. URL: http://content.sciendo.com/view/journals/mms/23/3/article-p359.xml (visited on 06/19/2019).

[164] A. Secerbegovic et al. “Blood pressure estimation using video plethysmography”. en. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. Prague, Czech Republic: IEEE, Apr. 2016, pp. 461–464. ISBN: 978-1-4799-2349-6. DOI: 10.1109/ISBI.2016.7493307. URL: http://ieeexplore.ieee.org/document/7493307/ (visited on 06/19/2019).

[165] Fabien Lotte and Cuntai Guan. “Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms”. In: *IEEE Transactions on biomedical Engineering* 58.2 (2010), pp. 355–362.

[166] Jonathan R Wolpaw et al. “Brain–computer interfaces for communication and control”. In: *Clinical Neurophysiology* 6 (2002). DOI: 10.1016/S1388-2457(02)00057-3.

[167] Gerwin Schalk and Eric C. Leuthardt. “Brain-Computer Interfaces Using Electrocorticographic Signals”. In: *IEEE Reviews in Biomedical Engineering* (2011). DOI: 10.1109/RBME.2011.2172408.

[168] Vikash Gilja et al. “Clinical translation of a high-performance neural prosthesis”. In: *Nature medicine* 21.10 (2015), pp. 1142–1145.

[169] Paul Nuyujukian et al. “Cortical control of a tablet computer by people with paralysis”. In: *PloS one* 13.11 (2018), e0204566.

[170] Brian N Pasley et al. “Reconstructing speech from human auditory cortex”. In: *PLoS biology* 10.1 (2012), e1001251.

[171] Mirco Ravanelli and Yoshua Bengio. “Speaker recognition from raw waveform with sincnet”. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 1021–1028.

[172] Morgan Stuart and Milos Manic. “Deep Learning Shared Bandpass Filters for Resource-Constrained Human Activity Recognition”. In: *IEEE Access* 9 (2021), pp. 39089–39097.

[173] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[174] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[175] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[176] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[177] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[178] Michael L. Waskom. “seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[179] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[180] Mirco Ravanelli, Titouan Parcollet, and Yoshua Bengio. “The Pytorch-kaldi Speech Recognition Toolkit”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ISSN: 1520-6149. May 2019, pp. 6465–6469. DOI: 10.1109/ICASSP.2019.8683713.

[181] Timothée Proix et al. “Imagined speech can be decoded from low- and cross-frequency intracranial EEG features”. In: *Nature Communications* 13 (2022), pp. 1–14.

[182] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. en. In: (Feb. 2015). URL: https://arxiv.org/abs/1502.03167v3 (visited on 09/06/2020).

[183] Jonathan Tompson et al. “Efficient object localization using Convolutional Networks”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 648–656.

[184] Jan Kubanek and Gerwin Schalk. "NeuralAct: a tool to visualize electrocortical (ECoG) activity on a three-dimensional model of the cortex". In: *Neuroinformatics* 13.2 (2015), pp. 167–174.

[185] Mark Patkowski. "Laterality effects in multilinguals during speech production under the concurrent task paradigm: Another test of the age of acquisition hypothesis". In: (2003).

[186] Chris Code. "Can the right hemisphere speak?" In: *Brain and Language* 57.1 (1997), pp. 38–59.

[187] Arthur S House et al. "Psychoacoustic speech tests: A modified rhyme test". In: *The Journal of the Acoustical Society of America* 35.11 (1963), pp. 1899–1899.

[188] M Ardussi Mines, Barbara F Hanson, and June E Shoup. "Frequency of occurrence of phonemes in conversational English". In: *Language and speech* 21.3 (1978), pp. 221–241.

[189] Emily M Mugler et al. "Direct classification of all American English phonemes using signals from functional speech motor cortex". en. In: *Journal of Neural Engineering* 11.3 (June 2014), p. 035015. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2560/11/3/035015. URL: https://iopscience.iop.org/article/10.1088/1741-2560/11/3/035015 (visited on 04/03/2023).

[190] G. Schalk et al. "BCI2000: a general-purpose brain-computer interface (BCI) system". In: *IEEE Transactions on Biomedical Engineering* 6 (2004). DOI: 10.1109/TBME.2004.827072.

[191] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[192] Christian Herff et al. "Generating Natural, Intelligible Speech From Brain Activity in Motor, Premotor, and Inferior Frontal Cortices". en. In: *Frontiers in Neuroscience* 13 (Nov. 2019), p. 1267. ISSN: 1662-453X. DOI: 10.3389/fnins.2019.01267. URL: https://www.frontiersin.org/article/10.3389/fnins.2019.01267/full (visited on 05/13/2020).

[193] Robin Tibor Schirrmeister et al. "Deep learning with convolutional neural networks for EEG decoding and visualization". In: *Human Brain Mapping* 38.11 (Nov. 2017), pp. 5391–5420. ISSN: 1065-9471. DOI: 10.1002/hbm.23730. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5655781/ (visited on 05/06/2019).

[194] Vasileios G Kanas et al. "Joint spatial-spectral feature space clustering for speech activity detection from ECoG signals". In: *IEEE Transactions on Biomedical Engineering* 61.4 (2014), pp. 1241–1250.

[195] Kristofer E. Bouchard et al. "Functional organization of human sensorimotor cortex for speech articulation". en. In: *Nature* 495.7441 (Mar. 2013), pp. 327–332. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature11911. URL: http://www.nature.com/articles/nature11911 (visited on 04/03/2023).

[196] Shreya Chakrabarti et al. "Progress in speech decoding from the electrocorticogram". In: *Biomedical Engineering Letters* 5 (Mar. 2015), pp. 10–21. DOI: 10.1007/s13534-015-0175-1.

[197] Christian Herff et al. "Brain-to-text: decoding spoken phrases from phone representations in the brain". en. In: *Frontiers in Neuroscience* 9 (June 2015). ISSN: 1662-453X. DOI: 10.3389/fnins.2015.00217. URL: http://journal.frontiersin.org/Article/10.3389/fnins.2015.00217/abstract (visited on 04/03/2023).

[198] Fabien Lotte et al. "Electrocorticographic representations of segmental features in continuous speech". en. In: *Frontiers in Human Neuroscience* 09 (Feb. 2015). ISSN: 1662-5161. DOI: 10.3389/fnhum.2015.00097. URL: http://journal.frontiersin.org/Article/10.3389/fnhum.2015.00097/abstract (visited on 04/03/2023).

[199] Josh Chartier et al. "Encoding of Articulatory Kinematic Trajectories in Human Speech Sensorimotor Cortex". en. In: *Neuron* 98.5 (June 2018), 1042–1054.e4. ISSN: 08966273. DOI: 10.1016/j.neuron.2018.04.031. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627318303398 (visited on 04/03/2023).

[200] Gopala K. Anumanchipalli, Josh Chartier, and Edward F. Chang. "Speech synthesis from neural decoding of spoken sentences". en. In: *Nature* 568.7753 (Apr. 2019), pp. 493–498. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-019-1119-1. URL: http://www.nature.com/articles/s41586-019-1119-1 (visited on 05/13/2020).

[201] Miguel Angrick et al. "Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity". In: *Communications Biology* 4.1 (2021), pp. 1–10.

[202] David A. Moses et al. "Real-time decoding of question-and-answer speech dialogue using human cortical activity". en. In: *Nature Communications* 10.1 (July 2019), p. 3096. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10994-4. URL: https://www.nature.com/articles/s41467-019-10994-4 (visited on 04/03/2023).

[203] David A. Moses et al. "Neuroprosthesis for Decoding Speech in a Paralyzed Person with Anarthria". en. In: *New England Journal of Medicine* 385.3 (July 2021), pp. 217–227. ISSN: 0028-4793, 1533-4406. DOI: 10.1056/NEJMoa2027540. URL: http://www.nejm.org/doi/10.1056/NEJMoa2027540 (visited on 04/03/2023).

[204] Stephanie Martin et al. "Word pair classification during imagined speech using direct brain recordings". In: *Scientific Reports* 6 (May 2016), p. 25803. DOI: 10.1038/srep25803.

[205] Stephanie Martin et al. "Decoding Inner Speech Using Electrocorticography: Progress and Challenges Toward a Speech Prosthesis". en. In: *Frontiers in Neuroscience* 12 (June 2018), p. 422. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00422. URL: https://www.frontiersin.org/article/10.3389/fnins.2018.00422/full (visited on 04/03/2023).

[206] Miguel Angrick et al. "Speech synthesis from ECoG using densely connected 3D convolutional neural networks". en. In: *Journal of Neural Engineering* 16.3 (June 2019), p. 036019. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/ab0c59. URL: https://iopscience.iop.org/article/10.1088/1741-2552/ab0c59 (visited on 05/13/2020).

[207] Joseph G. Makin, David A. Moses, and Edward F. Chang. "Machine translation of cortical activity to text with an encoder–decoder framework". en. In: *Nature Neuroscience* 23.4 (Apr. 2020), pp. 575–582. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-020-0608-8. URL: http://www.nature.com/articles/s41593-020-0608-8 (visited on 11/17/2022).

[208] Vasileios G Kanas et al. "Real-time voice activity detection for ECoG-based speech brain machine interfaces". In: *2014 19th International Conference on Digital Signal Processing*. IEEE. 2014, pp. 862–865.

[209] Kosuke Fukumori et al. "Epileptic Spike Detection Using Neural Networks With Linear-Phase Convolutions". In: *IEEE Journal of Biomedical and Health Informatics* 26.3 (2022), pp. 1045–1056. ISSN: 2168-2194, 2168-2208. DOI: 10.1109/JBHI.2021.3102247.

[210] Daniel Carey et al. "Functional and quantitative MRI mapping of somatomotor representations of human supralaryngeal vocal tract". In: *Cerebral Cortex* 27.1 (2017), pp. 265–278.

[211] Ulysse Côté-Allard et al. "Deep learning for electromyographic hand gesture signal classification using transfer learning". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.4 (2019), pp. 760–771.

[212] Mirco Ravanelli and Yoshua Bengio. "Speaker Recognition from Raw Waveform with SincNet". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. Dec. 2018, pp. 1021–1028. DOI: 10.1109/SLT.2018.8639585.

[213] Geoffrey E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". en. In: (July 2012). URL: https://arxiv.org/abs/1207.0580v1 (visited on 09/06/2020).

[214] Fergal Cotter. *Pytorch implementation of 2D Discrete Wavelet (DWT) and Dual Tree Complex Wavelet Transforms (DTCWT): fbcotter/pytorch_wavelets*. original-date: 2018-08-30T23:23:27Z. July 2019. URL: https://github.com/fbcotter/pytorch_wavelets (visited on 07/07/2019).

[215] Ulysse Côté-Allard. *MyoArmbandDataset*. https://github.com/UlysseCoteAllard/MyoArmbandDataset. 2020.

[216] Manfredo Atzori et al. "Building the Ninapro database: A resource for the biorobotics community". In: *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. ISSN: 2155-1782. June 2012, pp. 1258–1265. DOI: 10.1109/BioRob.2012.6290287.

[217] Angkoon Phinyomark and Erik Scheme. "EMG Pattern Recognition in the Era of Big Data and Deep Learning". en. In: *Big Data and Cognitive Computing* 2.3 (Sept. 2018), p. 21. DOI: 10.3390/bdcc2030021. URL: https://www.mdpi.com/2504-2289/2/3/21 (visited on 01/23/2020).

[218] Nadia Nasri et al. "Inferring Static Hand Poses from a Low-Cost Non-Intrusive sEMG Sensor". en. In: *Sensors* 19.2 (Jan. 2019), p. 371. ISSN: 1424-8220. DOI: 10.3390/s19020371. URL: http://www.mdpi.com/1424-8220/19/2/371 (visited on 02/07/2020).

[219] Muhammad Zia ur Rehman et al. "Multiday EMG-Based Classification of Hand Motions with Deep Learning Techniques". en. In: *Sensors* 18.8 (Aug. 2018), p. 2497. ISSN: 1424-8220. DOI: 10.3390/s18082497. URL: http://www.mdpi.com/1424-8220/18/8/2497 (visited on 01/13/2020).

[220] Tianzhe Bao et al. "Surface-EMG based Wrist Kinematics Estimation using Convolutional Neural Network". en. In: *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. Chicago, IL, USA: IEEE, May 2019, pp. 1–4. ISBN: 978-1-5386-7477-2. DOI: 10.1109/BSN.2019.8771100. URL: https://ieeexplore.ieee.org/document/8771100/ (visited on 01/13/2020).

[221] Na Duan et al. "Classification of multichannel surface-electromyography signals based on convolutional neural networks". en. In: *Journal of Industrial Information Integration* 15 (Sept. 2019), pp. 201–206. ISSN: 2452-414X. DOI: 10.1016/j.jii.2018.09.001. URL: http://www.sciencedirect.com/science/article/pii/S2452414X18300323 (visited on 02/06/2020).

[222] EH Rothauser. “IEEE recommended practice for speech quality measurements”. In: *IEEE Trans. on Audio and Electroacoustics* 17 (1969), pp. 225–246.

[223] Louis Collins. “3D Model-based segmentation of individual brain structures from magnetic resonance imaging data”. In: (1994).

[224] A.C. Evans et al. “3D Statistical Neuroanatomical Models from 305 MRI Volumes”. In: *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*. 1993, 1813–1817 vol.3. DOI: 10.1109/NSSMIC.1993.373602.

[225] Bruce Fischl. “FreeSurfer”. In: *NeuroImage* 62.2 (2012), pp. 774–781. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2012.01.021.

[226] Alexandre Gramfort et al. “MEG and EEG Data Analysis with MNE-Python”. In: *Frontiers in Neuroscience* 7.267 (2013), pp. 1–13. DOI: 10.3389/fnins.2013.00267.

[227] Martin Reuter et al. “Within-Subject Template Estimation for Unbiased Longitudinal Image Analysis”. In: *NeuroImage* 61.4 (2012), pp. 1402–1418. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2012.02.084.

[228] Wei-Ning Hsu et al. “Hubert: How Much Can a Bad Teacher Benefit ASR Pre-Training?” In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 6533–6537. DOI: 10.1109/ICASSP39728.2021.9414460.

[229] Alexei Baevski et al. “Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”. In: *arXiv:2006.11477 [cs, eess]* (2020). arXiv: 2006.11477 [cs, eess].

[230] Jonas Kohler et al. “Synthesizing Speech from Intracranial Depth Electrodes Using an Encoder-Decoder Framework”. In: *arXiv:2111.01457 [cs]* (2021). arXiv: 2111.01457 [cs].

[231] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: 10.48550/arXiv.1607.06450. arXiv: 1607.06450 [cs, stat].

[232] H Jégou, M Douze, and C Schmid. “Product Quantization for Nearest Neighbor Search”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1 (2011), pp. 117–128. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.57.

[233] Eric Jang, Shixiang Gu, and Ben Poole. *Categorical Reparameterization with Gumbel-Softmax*. 2017. DOI: 10.48550/arXiv.1611.01144. arXiv: 1611.01144 [cs, stat].

[234] Alexei Baevski, Steffen Schneider, and Michael Auli. “Vq-Wav2vec: Self-Supervised Learning of Discrete Speech Representations”. In: *arXiv:1910.05453 [cs]* (2020). arXiv: 1910.05453 [cs].

[235] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs].

[236] PZ Soroush et al. “Speech Activity Detection from Stereotactic EEG”. In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2021, pp. 3402–3407.

[237] Peter Langland-Hassan and Agustin Vicente. *Inner speech: New voices*. Oxford University Press, USA, 2018.

[238] Hanna S. Gauvin and Robert J. Hartsuiker. “Towards a New Model of Verbal Monitoring”. In: *Journal of Cognition* 3.1 (2020), p. 17. ISSN: 2514-4820. DOI: 10.5334/joc.81.

[239] Willem J. M. Levelt, Ardi Roelofs, and Antje S. Meyer. "A Theory of Lexical Access in Speech Production". In: *Behavioral and Brain Sciences* 22.1 (1999), pp. 1–38. ISSN: 1469-1825, 0140-525X. DOI: 10.1017/S0140525X99001776.

[240] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[241] Ashish Jaiswal et al. "A Survey on Contrastive Self-Supervised Learning". en. In: *Technologies* 9.1 (Dec. 2020), p. 2. ISSN: 2227-7080. DOI: 10.3390/technologies9010002. URL: https://www.mdpi.com/2227-7080/9/1/2 (visited on 07/04/2023).

[242] Yixin Liu et al. "Graph Self-Supervised Learning: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* (2022). arXiv:2103.00111 [cs], pp. 1–1. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2022.3172903. URL: http://arxiv.org/abs/2103.00111 (visited on 07/04/2023).

[243] Madeline C. Schiappa, Yogesh S. Rawat, and Mubarak Shah. "Self-Supervised Learning for Videos: A Survey". en. In: *ACM Computing Surveys* (Dec. 2022), p. 3577925. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3577925. URL: https://dl.acm.org/doi/10.1145/3577925 (visited on 07/04/2023).

[244] Shuo Liu et al. "Audio self-supervised learning: A survey". en. In: *Patterns* 3.12 (Dec. 2022), p. 100616. ISSN: 26663899. DOI: 10.1016/j.patter.2022.100616. URL: https://linkinghub.elsevier.com/retrieve/pii/S2666389922002410 (visited on 07/04/2023).

[245] Pranjal Kumar, Piyush Rawat, and Siddhartha Chauhan. "Contrastive self-supervised learning: review, progress, challenges and future research directions". en. In: *International Journal of Multimedia Information Retrieval* 11.4 (Dec. 2022), pp. 461–488. ISSN: 2192-6611, 2192-662X. DOI: 10.1007/s13735-022-00245-6. URL: https://link.springer.com/10.1007/s13735-022-00245-6 (visited on 07/04/2023).

[246] Xiao Liu et al. "Self-supervised Learning: Generative or Contrastive". en. In: *IEEE Transactions on Knowledge and Data Engineering* (2021). arXiv:2006.08218 [cs, stat], pp. 1–1. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2021.3090866. URL: http://arxiv.org/abs/2006.08218 (visited on 07/04/2023).

[247] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html (visited on 07/04/2023).

[248] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. In: *arXiv:1810.04805 [cs]* (May 2019). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 02/20/2022).

[249] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". en. In: *arXiv:2010.11929 [cs]* (June 2021). arXiv: 2010.11929. URL: http://arxiv.org/abs/2010.11929 (visited on 02/20/2022).

[250] Alexey Dosovitskiy et al. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv:2010.11929 [cs]* (2021). arXiv: 2010.11929 [cs].

[251] Wei-Ning Hsu et al. "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". In: *arXiv:2106.07447 [cs, eess]* (2021). arXiv: 2106.07447 [cs, eess].

[252] Yu-An Chung et al. "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training". In: *arXiv:2108.06209 [cs, eess]* (2021). arXiv: 2108.06209 [cs, eess].

[253] Timothée Proix et al. "Imagined Speech Can Be Decoded from Low- and Cross-Frequency Intracranial EEG Features". In: *Nature Communications* 13.1 (2022), p. 48. ISSN: 2041-1723. DOI: 10.1038/s41467-021-27725-3.

[254] Ardi Roelofs. "Spoken Word Planning, Comprehending, and Self-Monitoring: Evaluation of WEAVER++". In: *Phonological Encoding and Monitoring in Normal and Pathological Speech*. New York, NY, US: Psychology Press, 2005, pp. 42–63. ISBN: 978-1-84169-262-3.

[255] Shuji Komeiji et al. "Transformer-Based Estimation of Spoken Sentences Using Electrocorticography". In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 1311–1315. DOI: 10.1109/ICASSP43922.2022.9747443.

[256] Rothauser E. H. "IEEE Recommended Practice for Speech Quality Measurements". In: *IEEE Transactions on Audio and Electroacoustics* 17.3 (Sept. 1969). Publisher: Institute of Electrical and Electronics Engineers (IEEE), pp. 225–246. DOI: 10.1109/tau.1969.1162058. URL: https://cir.nii.ac.jp/crid/1361137045275638784 (visited on 07/01/2023).

[257] Xiaohua Zhai et al. *A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark*. arXiv:1910.04867 [cs, stat]. Feb. 2020. URL: http://arxiv.org/abs/1910.04867 (visited on 05/14/2023).

[258] Klemen Kotar et al. "Contrasting Contrastive Self-Supervised Representation Learning Pipelines". en. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9929–9939. ISBN: 978-1-66542-812-5. DOI: 10.1109/ICCV48922.2021.00980. URL: https://ieeexplore.ieee.org/document/9711402/ (visited on 05/14/2023).

[259] Simanto Saha et al. "Progress in Brain Computer Interface: Challenges and Opportunities". In: *Frontiers in Systems Neuroscience* 15 (Feb. 2021). DOI: 10.3389/fnsys.2021.578875.

[260] Clemens Brunner et al. "BNCI Horizon 2020: towards a roadmap for the BCI community". en. In: *Brain-Computer Interfaces* 2.1 (Jan. 2015), pp. 1–10. ISSN: 2326-263X, 2326-2621. DOI: 10.1080/2326263X.2015.1008956. URL: http://www.tandfonline.com/doi/full/10.1080/2326263X.2015.1008956 (visited on 04/03/2023).

[261] Miguel Angrick et al. "Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity". en. In: *Communications Biology* 4.1 (Sept. 2021), p. 1055. ISSN: 2399-3642. DOI: 10.1038/s42003-021-02578-0. URL: https://www.nature.com/articles/s42003-021-02578-0 (visited on 04/03/2023).

[262] Dongrui Wu et al. *Adversarial Attacks and Defenses in Physiological Computing: A Systematic Review*. arXiv:2102.02729 [cs]. Nov. 2022. URL: http://arxiv.org/abs/2102.02729 (visited on 06/02/2023).

[263] Tamara Denning, Yoky Matsuoka, and Tadayoshi Kohno. "Neurosecurity: security and privacy for neural devices". en. In: *Neurosurgical Focus* 27.1 (July 2009), E7. ISSN: 1092-0684. DOI: 10.3171/2009.4.FOCUS0985. URL: https://thejns.org/view/journals/neurosurg-focus/27/1/article-pE7.xml (visited on 03/09/2023).

[264] Sergio López Bernal et al. "Security in Brain-Computer Interfaces: State-of-the-art, opportunities, and future challenges". en. In: *ACM Computing Surveys* 54.1 (Jan. 2022). arXiv:1908.03536 [cs], pp. 1–35. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3427376. URL: http://arxiv.org/abs/1908.03536 (visited on 03/09/2023).

[265] Matthew Scholl et al. "An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule". en. In: ().

[266] Ron Ross et al. *Developing Cyber-Resilient Systems:: A Systems Security Engineering Approach*. en. Tech. rep. NIST SP 800-160v2r1. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2021, NIST SP 800–160v2r1. DOI: 10.6028/NIST.SP.800-160v2r1. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2r1.pdf (visited on 03/11/2023).

[267] Santiago Zanella-Béguelin et al. "Analyzing Information Leakage of Updates to Natural Language Models". en. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. arXiv:1912.07942 [cs, stat]. Oct. 2020, pp. 363–375. DOI: 10.1145/3372297.3417880. URL: http://arxiv.org/abs/1912.07942 (visited on 03/01/2023).

[268] Di Wu et al. "Understanding and defending against White-box membership inference attack in deep learning". en. In: *Knowledge-Based Systems* 259 (Jan. 2023), p. 110014. ISSN: 09507051. DOI: 10.1016/j.knosys.2022.110014. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705122011078 (visited on 03/03/2023).

[269] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. "Quantifying Privacy Leakage in Graph Embedding". en. In: *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. arXiv:2010.00906 [cs]. Dec. 2020, pp. 76–85. DOI: 10.1145/3448891.3448939. URL: http://arxiv.org/abs/2010.00906 (visited on 03/01/2023).

[270] Riccardo Presotto, Gabriele Civitarese, and Claudio Bettini. "Preliminary Results on Sensitive Data Leakage in Federated Human Activity Recognition". In: *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. Mar. 2022, pp. 304–309. DOI: 10.1109/PerComWorkshops53856.2022.9767215.

[271] Kambala Vijaya Kumar and Jonnadula Harikiran. "Privacy preserving human activity recognition framework using an optimized prediction algorithm". en. In: *IAES International Journal of Artificial Intelligence (IJ-AI)* 11.1 (Mar. 2022), p. 254. ISSN: 2252-8938, 2089-4872. DOI: 10.11591/ijai.v11.i1.pp254-264. URL: http://ijai.iaescore.com/index.php/IJAI/article/view/21134 (visited on 09/17/2023).

[272] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (visited on 10/24/2023).

[273] Eliza Strickland. "Andrew Ng, AI Minimalist: The Machine-Learning Pioneer Says Small is the New Big". In: *IEEE Spectrum* 59.4 (Apr. 2022). Conference Name: IEEE Spectrum, pp. 22–50. ISSN: 1939-9340. DOI: 10.1109/MSPEC.2022.9754503. URL: https://ieeexplore.ieee.org/abstract/document/9754503 (visited on 02/07/2024).